IL MIO COMMODORE 64

impariamo ad usarlo divertendoci con una raccolta di 50 utili e collaudati programmi e numerose note esplicative per modificarli a piacimento.

di ROGER VALENTINE

IL MIO COMMODORE 64

impariamo ad usarlo divertendoci con una raccolta di 50 utili e collaudati programmi e numerose note esplicative per modificarli a piacimento.

di ROGER VALENTINE

Traduzione a cura di Claudio Cerroni

Via dei Lavoratori, 124
CINISELLO BALSAMO (MI)

SOMMARIO

INTRODUZIONE

Benvenuti alla lettura di questo libro, uno dei migliori mai scritti per il Commodore 64, contenente ben 50 programmi pronti a partire.
Non sono tutti giochi, e questo scommetto che creera' del disappunto in qualcuno di voi, ma ve ne sono abbastanza da soddisfare anche i piu' esigenti.
Ci sono programmi per programmatori evoluti, illustranti la manipolazione di Bits e Bytes e la gestione dei Bits nell'uso di AND e OR.
Ci sono programmi per i principianti, e tra essi e' stato incluso un completo interprete PILOT, che permette la programmazione del 64 in un linguaggio ancora piu' semplice del BASIC.
Ci sono programmi che illustrano e facilitano l'uso degli sprites, dei caratteri ridefiniti e del suono.
Viene inoltre proposto il programma Cardbox, un completo ed esauriente Database, che da solo ha un valore commerciale superiore al prezzo di questa pubblicazione.
Sempre per i principianti c'e' un breve capitolo che introduce le semplici ed elementari caratteristiche del Commodore 64 (ad esempio PEEK e POKE), e per gli stravaganti c'e' il mio capitolo preferito (Interludio) che non contiene nemmeno un programma sensato.
Dopo tutto cio' potreste pensare che la parte principale di questo libro sia costituita dai listati; i programmi sono importanti, e' chiaro, ma spero che non li digitiate ciecamente (come se cio' fosse possibile) e vi troviate a pensare e a chiedervi come e perche' una routine esegue il suo lavoro.
Lo scopo principale di questo libro e', non dimenticatelo, mostrarvi come agiscono ed operano le routine utilizzate, e tutto cio' nel tentativo di insegnarvi cio' che e' essenziale circa la programmazione di un computer.
Ho aggiunto ad ogni programma delle note esplicative laddove le ritenevo necessarie, ed esse sono da leggere alla stessa stregua delle istruzioni.
Se non vi piacciono i programmi, o una parte di essi, ritenetevi pure liberi di apportare tutte le modifiche che ritenete necessarie.

Per far cio' dovrete leggere in modo critico i listati e le note di commento; in questo modo imparerete senza noia e facilmente il modo in cui i programmi dovevano essere scritti per soddisfare le vostre esigenze.

CINQUE PEZZI FACILI

..................................ASCII

Questo e' un semplicissimo quiz che giudichera' la vostra conoscenza del Codice ASCII, implementato, come sapete, nel vostro Commodore 64 (se non avete assolutamente idea di quello che sto parlando, andate a guardare la tabella F del manuale in dotazione).

Vengono usati tutti i caratteri che hanno un valore ASCII variante da 32 a 127; in questo modo sono eliminati tutti i tasti di controllo e molti, ma non tutti, caratteri grafici.
Un'intera fila di caratteri selezionati verra' visualizzata sullo schermo e voi dovrete indovinare l'appropriato valore ASCII.
Sarete costretti a rispondere correttamente prima di passare alla successiva domanda. Non e' prevista una fine delprogramma (ne potrete uscire in qualsiasi momento, come inogni altro programma di questo libro, premendo semplicemente il tasto RUN/STOP), e dopo ogni vostra risposta corretta compariranno sul video si ai secondi che avrete impiegato che il numero dei tentativi effettuato.

N.B.: se viene visualizzata una linea scura, la risposta e' 32 (barra spaziatrice). Non preoccupatevi se non riuscite subito a riconoscere i valori dei caratteri grafici (in ASCII dal 96 al 127), ma concentratevi sulla conoscenza di quelli alfa-numerici, specialmente quelli piu' comunemente usati, quali " (34), $ (36) e . (46).

..............................Note del programma

La linea 10, che non sembra avere una qualche importanza, ha invece un ruolo essenziale, come vi sara' spiegato piu'avanti nel Gioco dei dadi.

Le linee dalla 70 alla 150 sono un esempio di come si simula da programma una funzione quale INPUT.
La sostanziale differenza dall'uso del semplice INPUT consiste nella possibilita' di poter calcolare, grazie all'orologio del 64, il tempo di attesa della risposta.
Ma non e' tutto: digitando la seguente linea di programma:

```
75 NH=122 : NL=52 : GOSUB 1000
```

potrete aggiungere un fastidioso rumore di sottofondo che vi rammentera' l'urgenza della risposta (un ulteriore e pratica applicazione di questa routine "costruita", sebbene non utilizzabile in questo tipo di programma-quiz, consiste nella creazione di un valore massimo di tentativi per la variabile richiesta, e questo "tetto" deve essere considerato dal programma se il giocatore decide di non rispettare un predeterminato limite di tempo).

Le linee 80 e 90 costituiscono la parte principale della routine "costruita", che tiene conto del tasto RETURN (CHR$13), unico modo per uscire dal loop, e del tasto DELETE (CHR$ 20) per la correzione dell'errore.

TI$ e' l'orologio interno reale del Commodore 64, di cui si parlera' in maniera piu' estesa nel terzo capitolo.

```
5 REM *** PRG.#1 - ASCII ***
10 X=RND(-TI)
20 POKE53280,5:POKE53281,6:PRINTCHR$(147
);CHR$(144)
30 PRINT"■QUALE E' IL CODICE ASCII PER:-
■"
40 X=INT(RND(1)*96)+32
50 PRINT:FORJ=0TO19:PRINTCHR$(X);CHR$(32
);:NEXT
60 TI$="000000":F=1700:S=1
70 GETI$
80 IFI$=CHR$(13)THEN200
90 IFI$=CHR$(20)ANDL>0THENA$=LEFT$(A$,L-
1):GOTO120
100 IFI$<"0"ORI$>"9"THEN70
110 A$=A$+I$
120 PRINTCHR$(19):PRINT:PRINT:PRINT
130 PRINTTAB(18);A$;"  "
140 L=LEN(A$)
150 GOTO70
200 A=VAL(A$)
210 IFA=XTHEN300
220 NH=INT(F/256):NL=F-NH*256:GOSUB1000
230 F=F+200:IFF>13000THENF=1700
240 S=S+1
250 A$=""
260 GOTO120
300 T$=TI$
310 T=3600*VAL(LEFT$(T$,2))+60*VAL(MID$(
T$,3,2))+VAL(RIGHT$(T$,2))
330 PRINT:PRINT:PRINT:PRINT:PRINT"■■RISPO
STA ESATTA IN"T"SECONDI"
340 PRINT:PRINT"E"S"TENTATIVI";
350 IFS=1THENPRINT"■■O";
360 PRINT:PRINT:PRINT"■■GIOCHI ANCORA (S
/N)"
370 GETI$:IFI$="S"THENRUN20
380 IFI$<>"N"THEN370
390 END
1000 POKE54296,15
1010 POKE54277,15
1020 POKE54278,15
1030 POKE54273,NH
1040 POKE54272,NL
1050 POKE54276,33
1060 FORJ=1TO100:NEXT
1080 POKE54276,0
1090 POKE54277,0
1100 POKE54278,0
1110 RETURN
```

....PEEK & POKE

Non sarebbe bello se l'unico codice di valori riconosciutodal Commodore 64 fosse quello ASCII?
Purtroppo ogni carattere, oltre ad avere un valore ASCII possiede anche un valore nel codice PEEK e POKE (per saperne di piu', riguardatevi la tabella E del manuale d'uso in dotazione).

Il codice PEEK e POKE assume gli stessi valori di quello ASCII dal 32 al 63, ma per il resto sono completamente differenti. Inoltre i valori PEEK e POKE dal 128 al 255 rappresentano l'inverso dei caratteri dallo 0 al 127 (i caratteri inversi non sono contemplati nell'uso del set dei caratteri ASCII).

Questo programma mette alla prova la vostra conoscenza del codice PEEK e POKE, e include diverse raffinatezze non viste nel precedente programma.
Un singolo esempio del carattere proposto e' POKEato sullo schermo ed e' circondato da un quadrato rosso, cosi' da rendervi possibile un facile riconoscimento anche di quei caratteri grafici che sono molto simili tra loro (con un po' di pratica, naturalmente).
Avete solo sei possibilita' di rispondere correttamente col giusto valore prima che la giusta risposta compaia sullo schermo, e per ogni vostra risposta sbagliata vedrete il carattere corrispondente.

Dato che il carattere sara' sempre rappresentato in nero (colore del simbolo) e bianco (colore dello sfondo) sara' particolarmente facile riconoscere i caratteri inversi, quelli per intenderci con un codice superiore a 127.

N.B. : come per il programma precedente, e' possibile passare dal set dei carattere Testo a quello dei caratteri grafici con la semplice pressione dei tasti SHIFT ed il tasto Commodore.
Alcuni codici PEEK e POKE ed alcuni ASCII vengono rappresentati nello stesso modo, altri invece no. Sono spiacente se la cosa vi apparira' piu' complessa di quella che e' in realta' ma se non lo sapete il Commodore 64 possiede ben QUATTRO set di caratteri differenti che voi dovrete imparare!

..................Note del programma

Il POKEare i caratteri sullo schermo e' una tecnica ampiamente usata in questo libro e questa e' una delle

ragioni per le quali e' importante imparare l'uso di questi valori.
La prima versione del Commodore 64 costringeva chi voleva visualizzare un carattere sullo schermo ad usare oltre al valore dell'indirizzo (variante da 1024 a 2023) anche il valore del colore-sfondo (da 55296 a 56295).
Nelle versioni piu' recenti non si e' piu' riscontrata questa necessita'; il listato prevede questa eventualita' alle linee 60, 70 e 130.

Le due importanti formule (ma vi ricordo, per qualcuno di voi solo una, la prima) sono:

Indirizzo di schermo: 1024 + X + 40 * Y
(dove X e' la colonna verticale variante da 0 a 39, e Y e'la linea orizzontale da 0 a 24) e

Indirizzo Colore-video = Indir.di schermo + 54272

```
5 REM *** PRG.#2 - PEEK & POKE ***
10 POKE53280,10:POKE53281,1:PRINTCHR$(14
7);CHR$(144)
20 X=RND(-TI)
30 X=INT(RND(1)*256)
40 PRINT"QUALE E' IL CODICE PEEK/POKE PE
R:-"
50 FORJ=0TO2:FORK=0TO2
60 POKE1203+J+40*K,160:POKE55475+J+40*K,
2
70 NEXTK,J:POKE1244,X:POKE55516,0
80 FORJ=1TO8:PRINT:NEXT
90 S=S+1
100 PRINT"TENTATIVO #"S;TAB(10);:INPUTA$
110 A=VAL(A$):IFA>255ORA<0ORA<>INT(A)THE
NPRINTCHR$(145);:GOTO100
120 IFA=XTHENPRINT:PRINT"ESATTO":GOTO160
130 POKE1408+40*S,A:POKE55680+40*S,0
140 IFS<6THEN90
150 PRINT:PRINT:PRINT"IL CODICE ESATTO E
'"X
160 PRINT:PRINT"VUOI GIOCARE ANCORA (S/N
)"
170 GETA$:IFA$="S"THENRUN
180 IFA$<>"N"THEN170
```

..................POKE 53281: Colore dello sfondo

Sin dalle prime ore d'uso del vostro Computer, i numeri 53280 e 53281 dovranno imprimersi nella vostra memoria; questo perche' tramite loro potrete cambiare

rispettivamente il colore del bordo-video e dello sfondo. I valori da inserire tramite il comando POKE sono listati in fondo a questa pagina, e dovranno essere al piu' presto memorizzati da voi.
Per aiutare il processo di apprendimento, ecco qui un altro test: il bordo assume un colore casuale grazie alla funzione RANDOM e voi dovrete nel piu' breve tempo possibile premere il valore corrispondente in modo tale da colorare allo stesso modo anche lo sfondo.
Per aumentare un po' la difficolta' di questo esercizio ho pensato bene di usare ii sistema esadecimale (ad esempio i numeri da 10 a 15 corrispondono alle lettere dalla A alla F), ed in questo modo, se il bordo sara' blu dovrete premere il tasto E affinche' anche lo sfondo diventi blu.

Qualsiasi tasto voi premiate, limitatamente alle lettere riconosciute dal sistema esadecimale, il vostro video cambiera' colore fino a quando, in una corsa contro il tempo, non inserirete gli esatti venti valori utili a terminare il gioco.

..........................Note del programma

Penso che questa prova sia piu' facile delle due precedenti (ci sono infatti solo sedici valori da ricordare, contro un totale di 1024 locazioni di carattere diverse), cosi' la linea 120 conterra' solo la funzione VAL(TI$) per determinare il tempo in secondi, determinato nel momento in cui viene premuto il tasto esatto.
Non dovete farvi ingannare dal valore di TI$: infatti se dovesse comparire TI$= 000100, significhera' che avrete impiegato un minuto (e non cento secondi) per azzeccare il colore esatto.
Se riuscite a premere piu'di 16 tasti per arrivare alla soluzione, e' chiaro che vi meritereste una penalita'!
Comunque ricordatevi che l'uso di VAL(TI$) non e' il modo esatto per calcolare il tempo in secondi (per vedere il modo piu' giusto, andate a riguardare il programma sui valori ASCII).

CODICE DEI COLORI

Decimale	Esadecimale	Colore	CHR$(come confronto)
0	0	Nero	144
1	1	Bianco	5
2	2	Rosso	28
3	3	Azzurro	159
4	4	Porpora	156
5	5	Verde	30
6	6	Blu	31
7	7	Giallo	158
8	8	Arancione	129
9	9	Marrone	149
10	A	Rosso chiaro	150
11	B	Grigio	151
12	C	Grigio medio	152
13	D	Verde chiaro	153
14	E	Blu chiaro	154
15	F	Grigiochiaro	155

```
5 REM *** PRG. #3 - POKE 53281 ***
10 POKE53281,1:PRINTCHR$(147);CHR$(144)"
INDOVINA IL NUMERO DEL BORDO (1/0 - A/F)
"
20 X=RND(-TI):DIMT(20)
25 INPUT"[illegible]QUANTE GIOCATE";A:PRINT"[illegible]
                                        "
30 FORJ=1TOA
40 X=INT(RND(1)*16):IFX=YTHEN40
50 IFJ=1ANDX=1THEN40
60 POKE53280,X
70 TI$="000000"
80 GETI$:IFI$<"0"ORI$>"F"THEN80
90 A=ASC(I$)-48:IFA>16THENA=A-7
100 POKE53281,A
110 IFA<>XTHEN80
120 T=T+VAL(TI$):T(J)=T
130 C=5:IFX=1THENC=144
140 PRINTCHR$(147);CHR$(C)
150 FORK=1TOJ:PRINTK,"ESATTO IN"T(K),"SE
CONDI":NEXT
160 FORK=1TO1000:NEXTK
170 Y=X
180 NEXTJ
```

.....................Il gioco dei dadi

Voi sapete, naturalmente, che la funzione RND del Commodore 64 produce un numero casuale ed e' spesso usata nella forma N=INT(RND(1))*X)+1 per generare un numero

intero casuale da 1 a X. Ma cio' e' vero solo se diamo una interpretazione libera alla parola "casuale". Infatti quello che fa la funzione RND e' generare una lunga ma predefinita serie di valori che non hanno apparente relazione tra di loro. In altre parole, i numeri sembrano casuali, ma senoi conosciamo l'esatta posizione all'interno della sequenza l'intera serie e' prevedibile.

Quando voi accendete il vostro Commodore 64 il punto di partenza della sequenza assume un determinato valore (la STESSA posizione ogni volta!), cosi' ogni volta che chiameremo in causa la funzione RND verra' generato un numero che potra' essere riprodotto spegnendo e riaccendendo il computer.
Il "gioco dei dadi" utilizza questa caratteristica per presentare una apparentemente equa competizione tra due giocatori che si sfidano in dieci turni, proclamando vincitore colui che alla fine detiene il punteggio piu' alto. Il programma e' prevedibile anche se giusto: provate infatti a digitare il listato ed a salvarlo su nastro o su disco, e spegnete e riaccendete il computer prima di ricaricare il programma; se voi prenderete nota dei valori che appariranno ogni volta sullo schermo stabilirete la sequenza che il computer rispetta.
Provate tutte le volte che volete (spegnendo, riaccendendo il computer e ricaricando il programma) se non ci credete! La serie sara' sempre identica. (a questo proposito,quando giocherete contro qualche vostro amico, vi prego di ricordarvi di mandarmi, presso il mio editore, il 10% delle vincite...)

........................Note del programma

E' facile capire quanto sia ingiusto questo gioco dei dadi e quali svantaggi presenti l'uso della funzione RND quando viene usata in questo tipo di giochi.
Fortunatamente questo ostacolo puo' essere facilmente superato. Il numero 1 nell'espressione RND sopra menzionata (e alla linea 100 del programma) e' conosciuto come "valore fantoccio", perche' l'espressione RND deve essere seguita da un numero tra parentesi (un

"argomento"), ma il suo valore e' irrilevante, basta che sia POSITIVO.
Comunque, se l'argomento e' NEGATIVO non e' piu' un valore fantoccio. Infatti un argomento NEGATIVO viene usato per ri-inizializzare la sequenza dei numeri RANDOM.
Ad esempio, essa riporta il numero casuale ad uno specifico valore prè-determinato. Per far cio', provate a digitare la seguente linea:

```
5 X=RND(-1)
```

Il gioco rimarra' "ingiusto" ma almeno non dovrete spegnere e riaccendere il computer per ri-inizializzare il gioco.

Tutto cio' non risolve il nostro problema, naturalmente il sapere che e' possibile "resettare" la sequenza dei numeri casuali usando un argomento negativo ci permette di farlo tramite un valore sconosciuto. Convenzionalmente, e per fare le cose bene, il migliore modo per fare cio' e' usare la variabile TI propria del Commodore.
TI e' una variabile numerica dipendente dal valore TI$ delclock interno al computer, che si incrementa di un'unita' ogni sessantesimo di minuto dal momento in cui l'apparecchio viene acceso. E' umanamente impossibile per qualsiasi essere prevedere il valore di TI in un dato momento: cosi' digitiamo

```
5 X=RND(-TI)
```

```
5 REM *** PRG.#4 - GIOCO DADI ***
10 POKE53280,13:POKE53281,13:PRINTCHR$(1
47);CHR$(144);
20 FORJ=1TO8:B$=B$+CHR$(32):L$=L$+CHR$(2
26):NEXT
30 PRINT"GIOCAT.1",,"GIOCAT.2"
40 FORJ=1TO10:FORP=0TO1
50 PRINTCHR$(19);CHR$(17);
60 PRINTTAB(20*P);L$;CHR$(145)
70 PRINTTAB(20*ABS(P=0));B$;
80 PRINTCHR$(19):FORK=0TOJ:PRINTCHR$(17)
;:NEXT
90 GETI$:IFI$=""THEN90
100 X=INT(RND(1)*6)+1
110 PRINTTAB(20*P+2);X
120 T(P)=T(P)+X
130 PRINTCHR$(19):FORK=1TO7:PRINTCHR$(17
):NEXT
```

```
140 PRINTTAB(1)"TOTAL"TAB(21)"TOTALE"
150 PRINTTAB(20*P+2);T(P)
160 FORK=1TO500:NEXT
170 NEXTP,J
180 PRINT:PRINT:PRINT"FINE DEL GIOCO"
190 IFT(0)=T(1)THENPRINT"PARI":END
200 PRINT"IL GIOCAT."(T(0)>T(1))+2"VINCE
"
```

.......................Cursore e colore

Dopo una breve ed interessante escursione nel mondo del gioco d'azzardo tramite computer, torniamo ad un argomento che abbiamo gia' trattato: la gestione di PEEK e POKE sullo schermo.

Questo e' un programma che aiuta visivamente il programmatore (ci saranno altri esempi nel capitolo sette) e che illustra quali indirizzi di schermo e di codice-colore possono essere utilizzati. Avete a disposizione un cursore blu che dovrete muovere lungo tutto lo schermo grazie al consueto uso dei tasti di gestione-cursore (nella tastiera in basso a destra). Ad ogni movimento comparira' il valore corrispondente all'indirizzo interessato nell'angolo in alto a sinistra dello schermo.

A rendere il programma piu' completo viene inoltre illustrato l'uso dei comandi PRINTCHR$ e POKE646 che permettono di cambiare il colore del carattere visualizzato.
Il colore del testo sullo schermo puo' essere cambiato nel solito modo (usando i numeri da 1 a 8 contemporaneamente al tasto CTRL o al tasto Commodore). Non appena cambierete i colori del testo compariranno sul video sia i valori in codice ASCII (da usare nella forma PRINT CHR$ (num.)) sia il numero del codice-colore (variante da 0 a 15).

N.B. : dato che il colore dello sfondo in questo programma e' bianco, premendo i tasti CTRL+2 il testo risultera' invisibile.

La POKE 646 non e' un'espressione a voi molto familiare. Questo indirizzo contiene il codice di stampa del colore-carattere, cosi' come l'indirizzo 53280 il colore del bordo e 53281 il colore dello sfondo.

Il programmatore e' libero di scegliere se a questo scopo gli conviene usare la POKE 646 o il comando PRINT CHR$ seguito dal codice ASCII, o ancora la semplice parola PRINT seguita da un numero + il tasto CTRL o quello Commodore.

Le seguenti tre espressioni ottengono l'effetto di cambiare il colore di stampa in rosso:

```
POKE 646,2

PRINT CHR$ (28)

PRINT" [CTRL e il tasto 3]"
```

Generalmente, cosi' come nella stesura dei listati di questo libro, io preferisco usare il secondo metodo, ma la scelta e' tutta vostra.

La PEEK (653) alla linea 100 si riferisce all'indirizzo contenente un "flag" che indica lo stato dei tasti CTRL, SHIFT e Commodore.

```
PEEK (653) = 1 significa che il tasto SHIFT e' premuto
PEEK (653) = 2 significa che il tasto Comm. e' premuto
PEEK (653) = 4 significa che il tasto CTRL  e' premuto
```

E' da notare che il programma si accorge che state premendo uno di questi tasti abbinati ad un numero per cambiare il colore. La pressione di altri tasti lascia indifferente il programma; il tasto Commodore visualizza i caratteri semi-grafici presenti sulla sinistra di ogni tasto, ed il tasto CTRL produce una varieta' di effetti diversi tra i quali anche quelli provocati dai tasti-cursore. Se sullo schermo compariranno dei caratteri

non desiderati, essi potranno facilmente essere cancellati col successivo movimento del cursore o cambiando il colore di stampa (dalla linea 50, nella quale il comando CHR$(147) pulisce completamente lo schermo).

```
5 REM ** PRG. #5 - CURSORE & COLORE ***
10 POKE53280,9:POKE53281,1
20 A=30:C=5:S=1024:D=54272
30 POKE646,C
50 PRINTCHR$(147):PRINT"POSIZIONE SCHERM
O","POSIZIONE COLORE"
60 PRINTS,,S+D
70 PRINT:PRINT"PRINT COLORE <POKE>","<CH
R$>"
80 PRINTPEEK(646),,A
90 POKES,PEEK(S)+128:POKES+D,6
100 GETI$:P=PEEK(653)
110 IFP=2ORP=4THENPRINTI$;:IFPEEK(646)<>
CTHENC=PEEK(646):A=ASC(I$):GOTO50
120 N=N+40*((I$=CHR$(17))*(S<1983))-40*(
(I$=CHR$(145))*(S>1064))
130 N=S+((I$=CHR$(29))*(S<2023))-((I$=CH
R$(157))*(S>1024))
140 N=N+40*((I$=CHR$(17))*(S<1984))-40*(
(I$=CHR$(145))*(S>1063))
150 IFI$=CHR$(19)THENN=1024
160 IFS=NTHEN100
170 S=N:GOTO50
```

Giocaci Sam

............Comunita'

............Campo minato

............Chiudi la scatola

............Batticuore

............Catturato!

..............Comunita'

Avrete senz'altro sentito parlare di quei giochi che vengono definiti di "simulazione", dove grazie al computer si riesce ad essere ora regnanti di uno sterminato impero, ora capitani di un vascello intergalattico, ora al comando di una grande industria. In questo gioco io vi ho dato un

ruolo piu' modesto: siete a capo di una comunita' agricola auto-sufficiente, avete quattro amici, un acre di terra e un sacco di grano. Da queste umili origini dovete cercare di sviluppare la vostra comunita' fino ad arrivare ad avere una popolazione simile a quella di New York!
Ogni anno dovrete decidere quanto grano usare per la semina e quanto come nutrimento, quanto venderne e quanto conservarne per il futuro.
I parametri che vi aiuteranno nelle vostre decisioni sono i seguenti:

PIANTE: Ogni persona puo' piantare fino a due sacchi di grano ogni anno. Ogni acre di terra puo' contenere fino ad otto sacchi di semi. Se viene coltivato meno del 75% dell'acre, il 25% del terreno e' da considerarsi perso come punizione per la pigrizia dimostrata. Il raccolto ottenuto potra' essere povero, discreto o ricco, permettendovi di ottenere fino a due, quattro o sei volte quello che avevate piantato in precedenza.

CIBO: Ogni persona ha bisogno per nutrirsi di quattro sacchi di grano all'anno. Se voi provvederete a questo bisogno con una quantita' maggiore di quella necessaria, la popolazione si sviluppera' fino a consumare tutto il cibo in eccesso. Se invece fornirete meno cibo di quello richiesto la parte della comunita' che non ricevera' i quattro sacchi annuali morira'; in questo caso verrete accusati di cattiva gestione e dovrete subire il giudizio di una giuria di dodici persone. Se la maggioranza votera' per la vostra colpevolezza (ad esempio 10 contro 2) temo che il gioco sia destinato ad una fine prematura ma se la giuria vi riterra' non colpevole avrete a disposizione un altro anno per dimostrare le vostre qualita' di conduttore.

VENDITA: Dato che la vostra e' una comunita' auto-sufficiente non avete problemi legati ai soldi, cosi' cambierete immediatamente tutto il grano che vi e' avanzato con dell'altra terra; sfortunatamente il valore del terreno non puo' essere stabilito da voi, cosi' potreste ricevere sia 0 (si', zero!), sia 1, sia 2 acri di

terra per ogni sacco di grano venduto.
(N.B.: voi non potete perdere (come penalita') o guadagnare terreno (grazie alla vendita dei sacchi di grano) nello stesso anno. Se avete coltivato poco non vi e' permesso di disporre del grano avanzato per ottenere altri acri).

CONSERVAZIONE: Nessuna decisione e' importante quanto questa che vi permettera' di assimilare il grano invece di piantarlo, di venderlo o di usarlo come cibo.
Nei primi anni non vi potrete permettere il lusso di immagazzinare nemmeno un solo chicco di grano ma se sotto la vostra direzione la comunita' diventera' solida e prospera sarete in grado di decidere se rischiare di espandere troppo velocemente la comunita' o se risparmiare del grano in funzione dei possibili sviluppi futuri.

Lo scopo ultimo del gioco consiste nel creare una comunita' nel modo piu' grande possibile in un arco di tempo di 25 anni; ma dato che non e' affatto semplice sopravvivere,un buon obbiettivo da perseguire e' quello di raggiungere i 25 anni di vita (ricordatevi che oltre a correre il rischio di essere giudicato colpevole per le vostre errate decisioni, rischiate anche di ridurre a zero la popolazione della vostra comunita', e cio' significa che sareste condannati anche voi a morte!).

Poiche' il programma fa largo uso di testi e' facile capire cio' che ad ogni punto succede semplicemente leggendo in modo critico il listato; e' per questo che ho riassunto le note del programma in una tabella che vi guidi attraverso i punti principali:

Linee 10-30	Definizione iniziale della comunita'
Linee 100-180	Fine dell'anno in corso
Linee 190-370	Domande e risposte per l'anno seguente
Linee 500-640	Decremento popolazione e processo
Linee 650-670	Aumento della popolazione
Linee 710-740	Diminuzione della terra a disposizione
Linee 750-800	Acquisto di ulteriore terreno

Linee 810-870 Raccolto
Linee 1000-1040 Sub-routine: risposta nulla
Linee 2000-2140 Fine programma

```
5 REM *** PRG.#6- COMUNITA' ***
10 P=5: REM POPOLAZIONE ALL'ORIGINE
20 C=25:REM SACCHI DI GRANO A DISPOSIZ.
30 A=1: REM ACRI DI TERRENO DISPONIBILI
40 X=RND(-TI)
100 GOSUB1020
110 PRINTCHR$(5);CHR$(147)
120 PRINT"ANNO",,Y
130 PRINTCHR$(149)
140 PRINT"POPOLAZIONE DELLA COMUNITA'",P
150 PRINT"AREA DELLA COMUNITA'",A"ACRI"
160 PRINT"FORNITURA GRANO",C"SACCHI"
170 PRINTCHR$(17);CHR$(30);CHR$(18);
180 FORJ=1TO40:PRINTCHR$(192);:NEXT:PRIN
TCHR$(146)
190 Y=Y+1
200 IFY>25ORP=0THEN2000
210 PRINTCHR$(144)"ANNO",,Y
220 PRINTCHR$(28)
230 MAX=C:M=8*A:IFM<MAXTHENMAX=M
240 M=2*P:IFM<MAXTHENMAX=M
250 PRINT"QUANTI SACCHI DI GRANO VUOI PI
ANTARE"
260 PRINT"<MAX="MAX">";:INPUTS
270 IFS>MAXTHENGOSUB1000:GOTO260
280 C=C-S
290 OPT=4*P
300 PRINT"QUANTI SONO PER CIBO"
310 PRINT"<OPT="OPT">";:INPUTF
320 IFF>CTHENGOSUB1000:GOTO310
330 C=C-F
340 PRINT"E QUANTI NE VUOI VENDERE"
350 PRINT"<MAX="C">";:INPUTN
360 IFN>CTHENGOSUB1000:GOTO350
370 C=C-N
500 X=INT(F/4-P)
510 IFX>=0THEN650
520 POKE53280,0:POKE53281,12
530 PRINTCHR$(144);CHR$(147)
540 D$="PERSONA":IFX<-1THEND$="PERSONE"
550 PRINT-X;D$" MORTE"
560 PRINT
570 IFP=-XTHENPRINT"<L' INTERA POPOLAZIO
NE DEL COMUNE DISTRUTTA>":GOTO2100
580 PRINT"SEI ACCUSATO DI MAL GOVERNO"
590 PRINTCHR$(5)
600 PRINT"IL VERDETTO E' "CHR$(28);:D=IN
T(RND(1)*13):FORJ=1TO500:NEXT
610 IFD<10THENPRINTCHR$(158)"NON "CHR$(5
);:POKE53281,10
620 PRINT"COLPEVOLE"CHR$(5);D":"12-D
630 IFD>9THEN2100
640 GOTO700
650 POKE53280,7:POKE53281,10
660 PRINTCHR$(5);CHR$(147)
670 PRINT"POPOLAZIONE CRESCE DI"X
700 P=P+X
710 IFS>=A*6THEN750
720 I=INT(A*25)/100
730 PRINT:PRINTCHR$(28)"AREA DIMINUITA D
I"I"ACRI"CHR$(5)
```

```
740 A=A-I:GOTO810
750 I=INT(RND(1)*3):C$=CHR$(28)+"TROPPO
ALTI"
760 IFI=1THENC$=CHR$(31)+"MEDI"
770 IFI=2THENC$=CHR$(158)+"BASSI"
780 PRINT:PRINT"I PREZZI DELLA TERRA SON
O "C$;CHR$(5)
790 I=N*I:IFN>0THENPRINTCHR$(31)"NUOVA T
ERRA COMPRATA "I"ACRI"CHR$(5)
800 A=A+I
810 I=INT(RND(1)*3)+1:C$=CHR$(28)+"POVER
RO"
820 IFI=2THENC$=CHR$(31)+"MEDIO"
830 IFI=3THENC$=CHR$(158)+"BUONO"
840 PRINT:PRINT"IL RACCOLTO ERA "C$;CHR$
(5)
850 I=2*S*I
860 PRINT"GRANO PRODOTTO"I"SACCHI"
870 C=C+I
880 PRINT:PRINT:PRINT
890 PRINT"PREMI SPACE PER UN ALTRO ANN
O"
900 GETI$:IFI$<>CHR$(32)THEN900
910 GOTO100
1000 FORJ=0TO15:POKE53280,J:POKE53281,15
-J:NEXT
1010 PRINTCHR$(145);CHR$(145)
1020 POKE53280,5
1030 POKE53281,13
1040 RETURN
2000 POKE53280,4
2010 POKE53281,4
2100 PRINTCHR$(158)"COMUN. SOPRAVVISSUTA
 PER"Y-1"ANNI"
2110 PRINT:PRINT"VUOI GIOCARE ANCORA?(S/
N)"
2120 GETI$
2130 IFI$="S"THENRUN
2140 IFI$<>"N"THEN2120
```

................................Campo Minato

Questo e' un semplice ma ingannevole gioco. A prima vista sembra richiedere solo prontezza di riflessi, ma in effetti occorre una discreta agilita' mentale per vincere.

L'obbiettivo del gioco e' far compiere ai vostri tre uomini 27 passi attraverso un campo minato usando il lancio di un dado per stabilire il numero di passi da compiere per ogni mossa.

Tutte le mine sono chiaramente contraddistinte da un carattere grafico (il diamante) e a parte le due mine prossime alla fine, sono tutte ugualmente distanziate.
Sebbene la fortuna giochi la sua parte, come in ogni gioco legato alla sorte, l'abilita' consiste nel determinare

come l'uomo si debba muovere per sfruttare al meglio ogni mossa. Nel gioco badate innanzitutto alla vostra posizione se non volete essere costretti in una situazione suicida.

Gli omini sono colorati in viola, verde e blu e i tasti di controllo da usare sono quelli corrispondenti ai colori e cioe' i numeri 5, 6 e 7.
Le linee rappresentano le zone di sicurezza mentre il quadrato simboleggia la casa di partenza.

........................Note del programma

In questo listato ho usato i caratteri grafici simboleggiati come in tastiera (vedi la linea 40). E' una scelta assolutamente personale se usare direttamente i simboli grafici o i corrispondenti valori del codice ASCII abbinati a CHR$.

Il programma usa degli Array bidimensionali per contenere le informazioni riguardanti ogni uomo. La prima dimensione si riferisce semplicemente al numero dell'uomo (varia da 1 a 3, lo 0 non e' considerato), mentre la seconda dimensione ha due elementi, 0 che indica la posizio-ne dell'uomo e 1 che indica il suo stato (1= vivo: puo' essere mosso, 4= morto o a casa: non puo' essere mosso).

```
5 REM ***PRG.#7- CAMPO MINATO***
10 POKE53280,7:POKE53281,7:PRINTCHR$(147
);CHR$(149)
20 S=1024:D=54272
30 PRINT"[illegible]";CHR$(32);
40 FORJ=1TO7:PRINT"---◆";:NEXT:PRINT"◆--
*":REM CARATTERE MINA
50 DIMA(3,1):FORJ=1TO3:A(J,0)=1:A(J,1)=J
:
60 POKES+1+40*J,160:POKES+D+1+40*J,J+3:N
EXT
70 FORJ=1TO15:B$=B$+CHR$(32):NEXT
80 FORJ=1TO15:B$=B$+CHR$(157):NEXT
90 X=RND(-TI)
100 X=INT(RND(1)*6)+1
110 PRINT"[illegible]MUOVI"B$;X"PASSO";B
$;:IFX>1THENPRINT"■■"
120 PRINT"[illegible]QUALE UOMO MUOVI
(TASTI 5-7)";
```

```
130  GETM$:IFM$<"5"ORM$>"7"THEN130
140  M=VAL(M$)-4
150  IFA(M,1)>3THENGOSUB1000:GOTO120
160  N=A(M,0)+X:IFN>32THENN=32
170  P=M+3:IFN/4=INT(N/4)ORN=29THENP=0:IF
N=32THENP=1:A(M,1)=4:H=H+1
180  POKES+A(M,0)+40*M,32
190  A(M,0)=N
200  POKES+A(M,0)+40*M,160
210  POKES+D+A(M,0)+40*M,P:IFP=0THENGOSUB
1100
220  IFH+E<>3THEN100
300  R$="GAME OVER."
310  IFE=1THENR$=R$+" 1 UOMO UCCISO"
320  IFE=2THENR$=R$+" 2 UOMINI UCCISI"
330  IFE=3THENR$="TUTTI GLI UOMINI MORTI.
"
340  IFH=3THENR$="CONGRATULAZIONI"
350  PRINT"■■■■■":PRINTTAB(20-LEN(R$)/2)R$
400  PRINT"■"TAB(7)"PREMI ■SPAZIO■ PER GI
OCARE ANCORA"
410  GETR$:IFR$<>CHR$(32)THEN410
420  RUN
1000  NH=4:NL=208:W=33:GOSUB2000
1010  PRINT"■■■■■■■■■■■■■■■■■■■■■■■■"TAB(12)"MO
SSA SBAGLIATA"
1020  FORJ=1TO300:NEXT
1030  PRINT"]"TAB(12)B$
1040  GOSUB2100
1050  RETURN
1100  NH=9:NL=159:W=129:GOSUB2000
1110  FORJ=1TO600:NEXT
1120  A(M,1)=4:E=E+1
1130  GOSUB2100
1140  RETURN
2000  POKE54296,15
2010  POKE54277,200
2020  POKE54278,220
2030  POKE54273,NH
2040  POKE54272,NL
2050  POKE54276,W
2060  RETURN
2100  POKE54276,0
2110  POKE54277,0
2120  POKE54278,0
2130  RETURN
```

..........................Chiudi la scatola

"Chiudi la scatola" e' uno dei giochi piu' popolari nei Caffe' di tutta Europa anche se io non ho mai visto un locale provvisto di un Commodore 64 per divertire i clienti. Ogni numero puo' giocare e una partita si svolge in una serie di rounds che se ben giocati, e questo spiega la popolarita' del gioco nei locali, possono divertire per ore intere. Se non avete avversari, potete sempre esibirvi in una lotta contro voi stessi, per cercare un record sempre migliore.

Per ogni giocatore al suo turno, un round procede come segue:

Il giocatore ha di fronte a se' nove scatole, ognuna contraddistinta da un numero (da 1 a 9); tira quindi un paio di dadi (come avrete gia' capito, il possessore di un Commodore 64 non ha bisogno di possedere dei dadi veri), e chiude tutte le scatole che vuole facendo attenzione che la somma dei numeri sulle scatole corrisponda al numero uscito dal lancio dei dadi (per esempio, se esce il 9, si potra' chiudere la scatola 9, o le scatole 7 e 2, o le scatole 6 e 3, e cosi' via). Il giocatore rilancia nuovamente i dadi fino a chiudere tutte le scatole o fino a quando si trova impossibilitato a chiuderne ulteriormente. Il punteggio di ogni round viene calcolato a seconda delle scatole rimaste aperte.

Tutte le scatole vengono riaperte e i dadi passano al giocatore successivo. Vincitore del gioco sara' colui che avra' ottenuto il punteggio piu' basso dopo aver sommato lo "score" di ogni round.

Il miglior punteggio ottenibile in ogni round e' naturalmente 0 ma e' raro che si possa ottenere frequentemente.
Il peggior punteggio e' invece 43, per il quale e' necessario che esca per due volte consecutive il numero 2 (infatti col primo due si chiude la seconda scatola, e con il secondo il gioco e' gia' finito per l'impossibilita' da parte del giocatore di continuare).

Il programma dirige da solo tutti i meccanismi del gioco. L'unica cosa che l'operatore deve fare e' inserire i numeri delle scatole da chiudere.

Se scegliete di chiudere piu' di una scatola per ogni lancio, non premete RETURN dopo ogni numero, ma battete tutti i numeri in una sola linea, in qualsiasi ordine, e poi premete RETURN. Le risposte sbagliate saranno automaticamente respinte. Se non sarete in grado di procedere oltre, premete semplicemente RETURN senza

immettere alcun numero.

Eccetto che per due circostanze particolari (come diremo piu' avanti) il programma non si accorge quando il giocatore non ha piu' mosse a sua disposizione. Potreste aggiungere voi una routine che controlli quando non e' piu' possibile chiudere piu' alcuna scatola.
Un possibile algoritmo potrebbe essere il seguente:

Il massimo numero di scatole che puo' essere chiuso ad ogni lancio di dadi e' 4 (1+2+3+4=10, 10+5=15, che e' troppo alto per un lancio di dadi). Quindi usate quattro cicli FOR...NEXT, ognuno che conti da 1 a 9 dove vengano saltati i numeri gia' usciti, in modo tale che se al termine la somma delle variabili dei quattro cicli e' uguale al lancio dei dadi, allora sia possibile effettuare la mossa.

La stessa routine potrebbe essere usata non solo per controllare la situazione del giocatore, ma anche per dare anche al computer la possibilita' di giocare intelligentemente (potreste inoltre includere una "strategia di decisione" che permetta una scelta ponderata quando sono possibili diverse combinazioni).

Le due circostanze in cui il computer e' in grado di capire quando un turno e' finito sono quando tutte le scatole sono chiuse (linea 360) o quando il punteggio dei dadi e' maggiore della somma dei valori delle scatole ancora aperte.

```
5 REM *** PRG.#8-COLPISCI LA SCAT.*
10 A=RND(-TI)
20 FORJ=1TO6:B$=B$+CHR$(32):NEXT
30 FORJ=1TO5:B$=B$+CHR$(157):NEXT
40 POKE53280,5:POKE53281,6
50 PRINTCHR$(147);CHR$(144)
60 PRINTTAB(45);"◣▩IL GIOCO DELLA SCATOL
A■"
70 PRINT:INPUT"QUANTI GIOCATORI";P
80 PRINT:INPUT"QUANTI ROUNDS PER GIOCATO
RE";R :PRINT"■I"
90 IFI$=CHR$(20)ANDL>0THENA$=LEFT$(A$,L-
1):GOTO120
100 FORJ=1TOR:FORK=1TOP
110 PRINTCHR$(147);
120 PRINTTAB(5)"■ROUND"J;TAB(28);"■GIOCA
TORE"K:PRINTTAB(28);"■SCORE"S(K)
130 PRINTTAB(7);
```

```
140 FORH=1TO9:B(H)=H:B(0)=B(0)+H:PRINTH;
:NEXT:PRINTCHR$(156)
150 PRINTTAB(8);
160 FORH=1TO9:PRINTCHR$(18);CHR$(32);CHR
$(146);CHR$(32);CHR$(32);:NEXT
170 G=0:PRINTCHR$(144)
180 G=G+1
190 A=INT(RND(1)*6)+1
200 B=INT(RND(1)*6)+1
210 C=A+B
220 PRINT"TIRO NUMERO:"G
230 PRINT:PRINT"TIRO DADI:"A;B"="B$;C
240 IFC>B(0)THEN500
250 C$="":PRINT"QUALE SCAT
OLA CHIUDI"B$;:INPUTC$
260 L=LEN(C$):IFL=0THEN500
270 N=0:FORH=1TOL
280 T$=MID$(C$,H,1)
290 IFT$<"1"ORT$>"9"THEN400
300 T=VAL(T$):IFB(T)=0THEN400
310 N=N+T:NEXT:IFN<>CTHEN400
320 B(0)=B(0)-N:FORH=1TOL
330 T=VAL(MID$(C$,H,1))
340 GOSUB900
350 NEXT
360 IFB(0)=0THEN500
370 GOTO180
380 IFI$<>"N"THEN370
390 END
400 NH=5:NL=103:W=33
410 GOSUB800
420 PRINT""TAB(18);CHR$(18)"ERRORE"
CHR$(146)
430 FORH=1TO300:NEXT
440 PRINT""TAB(18)B$
450 GOSUB860
460 GOTO250
500 PRINT"FINE ROUND:
SCORE"B(0):S(K)=S(K)+B(0):B(0)=0
510 PRINT:PRINT"IL TUO PUNTEGGIO TOTALE"
S(K)
520 PRINT:PRINT:PRINT:PRINTTAB(4)"PREM
I LO SPAZIO PER CONTINUARE"
530 GETC$:A=RND(1):IFC$<>CHR$(32)THEN530
540 NEXTK,J
600 PRINTCHR$(147)"GIOCATORE","PUNTI"
610 W=45*R+1:FORK=1TOP
620 PRINTK,S(K)
630 IFS(K)=WTHENW$=W$+" &"+STR$(K)
640 IFS(K)<WTHENW=S(K):W$="GIOCATORE"+ST
R$(K)
650 NEXT
660 PRINT:PRINT"IL VINCITORE E' "W$
670 END
800 POKE54296,15
810 POKE54277,200
820 POKE54278,250
830 POKE54273,NH
840 POKE54272,NL
850 POKE54276,W:RETURN
860 POKE54276,0
870 POKE54277,0
880 POKE54278,0:RETURN
900 NH=48:NL=127:W=17
910 GOSUB800
920 PRINT"";TAB(5);
930 FORM=1TOT:PRINT"";:NEXT
```

```
940 PRINTCHR$(156);CHR$(18);CHR$(32);CHR
$(146);CHR$(144);
950 PRINTSPC(9);CHR$(32)
960 B(T)=0
970 GOSUB860:RETURN
1000 POKE54296,15
1010 POKE54277,15
1020 POKE54278,15
1030 POKE54273,NH
1040 POKE54272,NL
1050 POKE54276,33
1060 FORJ=1TO100:NEXT
1080 POKE54276,0
1090 POKE54277,0
1100 POKE54278,0
1110 RETURN
```

...........................Batticuore

Un titolo abbastanza melodrammatico, questo, che sebbene non abbia alcuna intenzione di procurare qualche infarto riesce a creare momenti di reale tensione.

Non appena i Cuori precipiteranno dal cielo voi dovrete fare di tutto perche' non tocchino il suolo; avrete a disposizione una sorta di racchetta, il cui movimento dipendera' dalla pressione dei due tasti cursori (il tasto della qestione orizzontale permettera' lo spostamento a destra, mentre il tasto della gestione verticale permettera'lo spostamento verso sinistra); questa scelta sembrerebbe illogica, ma e' molto piu' veloce premere un solo tasto piuttosto che due contemporaneamente.

Il gioco finira' quando si formeranno otto colonne di quattro cuori ciascuno; il punteggio verra' aumentato di una unita' ad ogni secondo di sopravvivenza, mentre diminuira' di cinque per ogni cuore "sfracellato" al suolo.

.......................Note del programma

La PEEK (197) (alla linea 400) controlla quale tasto viene in ogni momento premuto. Ogni tasto influisce sul contenuto dell'indirizzo 197 in un unico modo, sebbene, sfortunatamente, il valore non corrisponde ne' al codice

ASCII ne' al codice PEEK/POKE. Comunque voi potete facilmente scoprire il valore di ogni tasto premuto digitando:

```
1 PRINT PEEK(197): GOTO 1
```

(PEEK(197) = 64 quando nessun tasto viene premuto)

Ai fini del gioco, i soli valori di PEEK(197) che hanno valore sono 7 (cursore alto/basso) e 2 (cursore destra/sinistra). Se preferite usare due differenti tasti per la gestione della racchetta (se siete mancini, per esempio), basta cambiare i valori presenti nella linea 410, dopo averne scoperti i valori relativi per fare in modo che il programma riconosca i tasti desiderati.

La linea 410 usa la logica Booleana per determinare la posizione successiva della racchetta. Usando questa logica, il Commodore 64 assegna il valore -1 all'affermazione VERO e 0 all'affermazione FALSO. Cosi' e' vero che P=7 quando il cursore-sinistra e' premuto, e X>1864 quando la posizione della racchetta non e' al margine estremo sinistro dello schermo. Se entrambi le affermazioni sono vere allora il computer interpretera' la linea 410 come segue:

```
N = X-((-1)*(-1))+((0)*(-1))
```

cioe'

```
N = X-1
```

in questo modo la racchetta si muovera' di un posto verso sinistra. P=7 e P=2 non possono essere vere allo stesso tempo, cosi' almeno uno dei due valori sara' sempre uguale a 0; e' possibile invece che entrambi le espressioni siano false (=0), quando cioe' nessun tasto viene premuto. X sara' uguale a N e la racchetta rimarra' ferma.

```
5 REM ***PRG# 9-BATTICUORE ***
10 X=RND(-TI)
20 POKE53280,6:POKE53281,3:PRINTCHR$(147
);CHR$(144);
30 X=1884:D=54272
40 TI$="000000"
50 PRINTCHR$(19):PRINT:PRINT"TEMPO "TI$,
"PENALITA'PUNTI"S
100 FORJ=0TO1:H=H(J)
110 IFH=0THENH(J)=INT(RND(1)*40)+1424:H=
H(J)
120 IFH>1983THENH(J)=0:GOTO190
130 Q=PEEK(H+40)
140 IFQ=120THENPOKEH,121:FORK=1TO50:NEXT
:POKEH,32:H(J)=0:GOTO190
150 IFQ=102THENH(J)=H-40:IFH<1904THEN500
160 C=83:IFQ=102ORH>1943THENC=102:S=S+5
170 POKEH,32:H(J)=H(J)+40:POKEH(J),C:POK
EH(J)+D,2
180 IFQ=102THENH(J)=0
190 GOSUB400:NEXT
200 GOTO50
400 P=PEEK(197)
410 N=X-((P=7)*(X>1864))+((P=2)*(X<1902)
)
420 IFN<>XTHENPOKEX-(P=7),32:X=N
430 POKEX,120:POKEX+1,120:POKEX+D,0:POKE
X+D+1,0
440 RETURN
500 T$=TI$
510 S=3600*VAL(LEFT$(T$,2))+60*VAL(MID$(
T$,3,2))+VAL(RIGHT$(T$,2))-S
520 PRINT:PRINT"IL TUO PUNTEGGIO E'",S"P
UNTI"
530 PRINT:PRINT"PREMI ▒SPAZIO■ PER GIOCA
RE ANCORA"
540 GETI$:IFI$<>CHR$(32)THEN540
550 RUN20
```

........................Catturato!

Questo gioco e' semplicissimo (come lo sono sempre del resto tutti i giochi migliori) ma molto crudele. Appena muoverete il carattere a vostra disposizione (un cuore) sullo schermo usando i tasti-cursore (che hanno stavolta le loro normale funzioni poiche' e' permesso anche il movimento verticale) una serie di Clubs compariranno per cercare di intrappolarvi in una morsa mortale. Comunque i Club non cercheranno di catturarvi ma solo di circondarvi; non sara' quindi difficile sopravvivere abbastanza a lungo alla caccia, a meno che non pensiate ad un eventuale suicidio...

Potete, prima dell'inizio del gioco, stabilire un handicap, decidere cioe' a vostro piacere quanti Clubs debbano essere gia' presenti sul campo da gioco; fate attenzione a non esagerare per non essere catturati ancor prima di cominciare!

..........................Note del programma

Il movimento del vostro carattere (linea 190) usa una logica simile a quella del gioco precedente. La differenza consiste nel fatto che invece di essere relegati su una unica linea di schermo, potrete muovervi in alto, in basso, a destra e a sinistra (linee 200-230).

Solitamente il posizionamento dei Clubs avviene su uno dei quattro punti cardinali adiacenti la vostra posizione (linee 120-130) oppure casualmente se ci troviamo all'inizio del gioco (subroutine alla linea 900).

Ho deliberatamente lasciato un paio di errori nel programma! Ci sono due modi per raggiungere un punteggio alto senza sfoggiare una particolare abilita'. Mi chiedo quanto tempo impiegherete a scoprirli (provate a correggere voi stessi gli errori, ma prima leggete i suggerimenti qui sotto).

.............Suggerimenti per i cacciatori d'errori

1 - Il cuore spazia per tutto lo schermo, ma come fanno i Clubs ?

2 - La linea 190 tiene conto di TUTTI i valori che puo' assumere A ?

```
5 REM *** PRG.# 10- CATTURATO! ***
10 POKE53280,3:POKE53281,1:PRINTCHR$(147
):CHR$(144)
20 L=1024:D=54272
30 H=83:C=88
40 X=RND(-TI)
50 S=-1:X=500
60 INPUT"QUANTI CLUBS PER INIZIARE(FORSE
 0)";Q:PRINTCHR$(147):IFQ=0THEN100
```

```
70 FORJ=1TOQ:GOSUB900
80 POKEB+D+L,0:POKEB+L,C:NEXT
100 POKEX+D+L,2:POKEX+L,H
110 NH=68:NL=149:GOSUB1000
120 B=INT(RND(1)*4)
130 B=X+(B=0)-(B=1)+40*(B=2)-40*(B=3)
140 IFB<40ORB>999THENGOSUB900
145 IFPEEK(B+L)=CTHENGOSUB900
150 POKEB+D+L,0:POKEB+L,C
160 NH=17:NL=37:GOSUB1000
165 S=S+1:PRINTCHR$(19);"SCORE"S
170 I$="":GETI$:IFI$=""THEN170
180 A=ASC(I$)
190 N=X+(A=157)-(A=29)+40*(A=145)-40*(A=
17)
200 IFX/40=INT(X/40)ANDN=X-1THENN=X+39
210 IFN/40=INT(N/40)ANDN=X+1THENN=X-39
220 IFN<40THENN=N+960
230 IFN>999THENN=N-960
240 POKEX+D+L,1:POKEX+L,32
250 X=N
260 IFPEEK(X+L)<>CTHEN100
300 NH=4:NL=73:GOSUB1000
310 POKEX+D+L,4:POKEX+L,86
320 NH=2:NL=37:GOSUB1000
330 PRINTCHR$(19);TAB(12)"PREMI  SPAZIO 
 PER RIGIOCARE"
340 GETI$:IFI$<>CHR$(32)THEN340
350 RUN
900 B=INT(RND(1)*960)+40
910 RETURN
1000 POKE54296,15
1010 POKE54277,15
1020 POKE54278,15
1030 POKE54273,NH
1040 POKE54272,NL
1050 POKE54276,33
1060 FORJ=1TO100:NEXT
1080 POKE54276,0
1090 POKE54277,0
1100 POKE54278,0
1110 RETURN
```

Giocando col tempo

.....................Orologio-sveglia

.....................Orologio-calendario

.....................Cronometro

.....................Orologio-sveglia

Abbiamo gia' visto come la variabile dedicata TI$ ci permetta di accedere all'orologio interno del Commodore

64. Molti computers possiedono questa caratteristica funzione ma la Commodore e' stata perticolarmente generosa nel provvedere in modo tale da avere non solo un timer ma un vero e proprio calcolatore di tempo reale, espresso in ore, minuti, secondi. Avendo a disposizione una tale comodita' sarebbe uno spreco non saperla sfruttare a dovere.

Per inizializzare l'allarme, se ne avete bisogno di uno, inserite l'ora ed i minuti esatti in cui volete che l'allarme suoni. Quindi regolate l'orologio inserendo le ore ed i minuti esatti (ricordatevi che e' un orologio che conta 24 ore); fatto questo, aspettate lo scoccare di un minuto per far partire i secondi tramite la semplice pressione della barra spaziatrice.

Quando l'allarme entrera' in funzione, accompagnato da un segnale luminoso sullo schermo, bastera' premere un tasto qualsiasi per spegnerlo.

N.B. : Una volta che l'allarme e' stato regolato (e cosi' pure l'orologio) non dovete spegnere il computer. Semmai potrete spegnere il monitor/televisore, o anche far girare altri programmi sul computer (badando pero' che essi non modifichino il valore di TI$). Se intendete ricaricare "orologio-sveglia" dopo aver fatto girare altri programmi e volete visualizzare il tempo reale senza resettare l'orologio, usate RUN 200 (in ogni caso andra' reinserito il momento desiderato per il suono dell'allarme).

......................Note del programma

Quando inizializzate l'orologio, e' importante capire che TI$ accetta solo valori validi, altrimenti si incorrera' in un "Illegal Quantity Error". Le linee 520, 550 e 560 si accertano che H$ (che mette in funzione TI$ alla linea 130) sia sempre formata da sei cifre in lunghezza, mentre le linee 510 e 540 controllano che i numeri siano nell'arco dei valori accettabili. Ricordatevi, quando trasformerete dei numeri in stringhe, che i numeri

positivi sono sempre preceduti da uno spazio (ad es. STR$(8) sara' " 8" e non "8". Per eliminare questo "spazio" usate MID$(STR$(8),2) nelle linee 520 e 550.

La linea 220 richiama la subroutine di allarme nel momento in cui TI$ e' uguale al predeterminato tempo T$. Questa subroutine viene chiamata una volta sola, ma il suono prodotto ha un valore di Sustain/Release di 240 (linea 320), che essendo il valore massimo permette un suono costante che continua fino allo spegnimento (linea 230).

```
5 REM ***PRG.# 11-OROL.SVEGLIA ***
10 POKE53280,1:POKE53281,1
20 PRINTCHR$(147);CHR$(144)
30 PRINT"INSERISCI L'ALLARME(S/N)?"
40 GETA$:IFA$="N"THENT$="D":GOTO100
50 IFA$<>"S"THEN40
60 GOSUB500
70 T$=H$
100 PRINTCHR$(147)
110 PRINT"PREGO, INSERISCI L'ORA ESATTA"
120 GOSUB500
130 TI$=H$
200 PRINTCHR$(147)
210 PRINT"[illegible]"TAB(15);LEFT$(TI
$,2)"."MID$(TI$,3,2)"."RIGHT$(TI$,2)
220 IFTI$=T$ANDA=0THENGOSUB300
230 GETI$:IFI$<>""THENGOSUB400
240 IFA=1THENGOSUB380
250 GOTO210
300 POKE54296,15
310 POKE54277,240
320 POKE54278,240
330 POKE54273,72
340 POKE54272,169
350 POKE54276,33
360 PRINTCHR$(19)"PREMI UN TASTO PER SPE
GNERE L'ALLARME"
370 A=1
380 POKE53280,INT(RND(1)*14)+2
390 POKE53281,INT(RND(1)*14)+2
395 RETURN
400 POKE54276,0
410 POKE54277,0
420 POKE54278,0
430 A=0
440 POKE53280,1
450 POKE53281,1
460 PRINTCHR$(19);:FORJ=1TO39:PRINTCHR$(
32);:NEXT
470 RETURN
500 INPUT"ORE (0-23)";H
510 IFH<0ORH>23ORH<>INT(H)THENPRINTCHR$(
145);:GOTO500
520 A$=MID$(STR$(H),2):IFLEN(A$)=1THENA$
="0"+A$
530 INPUT"MINUTI (0-59)";M
540 IFM<0ORM>59ORM<>INT(M)THENPRINTCHR$(
145);:GOTO530
```

```
550 B$=MID$(STR$(M),2):IFLEN(B$)=1THENB$
="0"+B$
560 H$=A$+B$+"00"
570 IFT$=""THENRETURN
580 PRINT"PREMI SPAZIO PER AVVIARE L'ORO
LOGIO"
590 GETI$:IFI$<>CHR$(32)THEN590
600 RETURN
```

.......................Orologio/calendario

Non sono molto sicuro che sia economicamente sensato usare il vostro Commodore 64 come un semplice calendario ma al mondo dovesse mai scarseggiare la carta, questo programma diventerebbe inestimabile. Esso ha infatti il vantaggio di essere un calendario perpetuo (valido "solo" fino al 9999), sebbene potreste avere qualche inesattezza alla ricerca di date storiche, perche' dovete tener conto del fatto che il moderno calendario (Gregoriano) non fu introdotto nella maggioranza dei paesi che dopo il 1582.

L'anno corrente e' il 1984, anno in cui ho scritto questo programma, ma se voi arriverete a leggere questo libro solo nel 1995 vi bastera' cambiare la linea 45. Voi potete in ogni caso richiedere un calendario per ogni anno ma, mentre per l'anno corrente bastera' premere un tasto per ogni altro anno occorrera' specificarlo completamente.

Il mese desiderato puo' essere richiamato specificando l'intera parola (ad es. Febbraio), o il suo corrispondente numero (ad es. 2) o un'abbreviazione del suo nome (ad es. febbr, feb o f).

Una volta che il calendario viene visualizzato, un menu' sottostante vi propone una serie di opzioni:

N e L : mostra il mese precedente o quello successivo
C e X : accende o spegne l'orologio
D : inserisce la data
B : cancella il testo del menu' (basta premere nuovamente il tasto per vederlo riapparire)

Alcune di queste opzioni lavorano congiuntamente, cosi' se per esempio l'orologio e' acceso ed il calendario e'

funzionante, la data si cambiera' automaticamente allo scoccare della mezzanotte. Allo stesso modo, giunti alla fine del mese, verra' visualizzato direttamente il mese successivo, indicante il primo giorno del mese.

```
3 REM ***PRG.# 12- OROL.CALENDARIO ***
5 POKE53280,6:POKE53281,1:PRINTCHR$(31)
10 DIMA(31)
15 N$=" 1 2 3 4 5 6 7 8 910111213141516171819202122232425262728293031"
20 M$="*GENNAIO*FEBBRAIO*MARZO*APRILE*MAGGIO*GIUGNO*LUGLIO*AGOSTO*SETTEMBRE*"
25 M$=M$+"OTTOBRE*NOVEMBRE*DICEMBRE*"
30 LL=LEN(M$)
35 D$="DOMLUNMARMERGIOVENSAB"
40 C$="303232332323"
45 TY=1984:REM *** QUEST'ANNO***
50 FORJ=1TO39:CL$=CL$+CHR$(32):NEXT
100 PRINTCHR$(147);TAB(15)"■CALENDARIO■"
110 PRINT"■■■■■■VUOI IL"TY" ? (S/N)"
120 GETQ$:IFQ$=""THEN120
130 IFQ$="S"THEN200
140 IFQ$<>"N"THEN120
150 PRINT
160 INPUT"ANNO RICHIESTO (4 CIFRE)       *■■■■";TY$
170 IFTY$="*"THENPRINT"■■■":GOTO160
180 IFLEN(TY$)<>4ORVAL(TY$)<1ORVAL(TY$)<>INT(VAL(TY$))THENPRINT"■■■":GOTO160
190 TY=VAL(TY$)
200 PRINT"■■■■■■■■■■■■■■"
205 INPUT"MESE RICHIESTO      *■■■■";TM$
210 IFTM$="*"THENPRINT"■■■":GOTO205
215 TM=VAL(TM$):IFTM>0ANDTM<13ANDTM=INT(TM)THEN300
220 IFTM>12THENTM$="*":GOTO210
225 L=LEN(TM$)
230 TM=0:FORJ=1TOLL-L
235 IFMID$(M$,J,1)="*"THENTM=TM+1:GOTO245
240 IFTM$=MID$(M$,J,L)THENJ=LL+2
245 NEXTJ:IFJ=LL+1THENTM$="*":GOTO210
250 B=0
300 D=0
305 X=0:FORJ=1TOLL
310 IFX=TMTHENTM$=MID$(M$,J):GOTO325
315 IFMID$(M$,J,1)="*"THENX=X+1
320 NEXT
325 FORJ=1TOLEN(TM$)
330 IFMID$(TM$,J,1)="*"THENTM$=LEFT$(TM$,J-1):GOTO340
335 NEXT
340 LY=0
345 IFTY/4<>INT(TY/4)THEN360
350 IFTY/100<>INT(TY/100)THENLY=1:GOTO360
355 IFTY/400=INT(TY/400)THENLY=1
360 TN$=N$
365 IFTM=9ORTM=4ORTM=6ORTM=11THENTN$=LEFT$(N$,60)
370 IFTM=2THENTN$=LEFT$(N$,56):IFLY=1THENTN$=LEFT$(N$,58)
400 PRINT"■ "TY;TAB(20-LEN(TM$)/2);"■";TM$;"■";TAB(33);TY
```

```
410  PRINT:PRINT:PRINT
420  FORJ=1TOLEN(D$)STEP3
430  PRINTMID$(D$,J,3)"▮"
440  NEXT
450  X=0
460  FORJ=1TOTM
470  X=X+ASC(MID$(C$,J,1))-20
480  NEXT
490  IFLY=1ANDTM>1THENX=X+1
500  X=X+(TY-1)*365+INT((TY-1)/4)-INT((TY
-1)/100)+INT((TY-1)/400)
510  X=X+1-LEN(TN$)/2
520  X=X-7*INT(X/7)
530  S$="▶▶▶▶▶"
540  PRINT"▮▮▮"S$
550  IFX=0THEN570
560  FORJ=0TOX-1:PRINT"▮"S$:NEXT
570  FORJ=1TOLEN(TN$)STEP2
580  X=X+1
590  IFX=8THENX=1:S$=S$+"▶▶▶▶▶":PRINT"▮▮▮
▮";
600  PRINT"▮"S$;
610  A(INT(J/2)+1)=1024+PEEK(211)+40*PEEK
(214)
620  PRINTMID$(TN$,J,2)
630  NEXT
640  IFD<>0THENGOSUB3060
650  IFBTHENGOSUB2000:GOTO700
660  PRINT"▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮"
670  PRINTCHR$(144)"N = MESE PRIMA      : L
 = MESE DOPO"
680  PRINT"▮C = AVVIA OROLOG.  : X = SPEGN
I OROLOG."
690  PRINT"▮D = INSERISCI DATA: B = CANCE
LLA TESTO";CHR$(31)
700  GETQ$
705  IFC=0THEN725
710  PRINT"▮▮▮▮▮▮▮▮▮▮▮▮▮▮▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶
▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶";
715  PRINT"▮"LEFT$(TI$,2)"."MID$(TI$,3,2)
"."MID$(TI$,5)"▮"
720  IFTI$="000000"ANDD<>0THEN4000
725  IFQ$=""THEN700
730  IFQ$<>"N"ANDQ$<>"L"THEN760
735  IFQ$="N"THENTM=TM+1:IFTM>12THENTM=1:
TY=TY+1
740  IFTY>9999THENTY=9999:TM=12:GOTO700
745  IFQ$="L"THENTM=TM-1:IFTM<1THENTM=12:
TY=TY-1
750  IFTY<1THENTY=1:TM=1:GOTO700
755  GOTO300
760  IFQ$="B"THENB=NOTB:GOTO650
765  IFQ$="C"THENGOSUB900:GOTO650
770  IFQ$="X"THENC=0:PRINT"▮▮▮▮▮▮▮▮▮▮▮▮▶▶
▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶          "
775  IFQ$="D"THENGOSUB3060:GOSUB3000:GOTO
650
780  GOTO700
900  GOSUB2000
910  C=1
920  PRINT"▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮"
930  INPUT"ORA          *▮▮▮";HT$
940  A=VAL(HT$):L=LEN(HT$)
950  IFA=0ANDHT$<>"0"ANDHT$<>"00"THENPRIN
T"▮▮":GOTO930
960  IFL>2ORA<0ORA>23ORA<>INT(A)THENPRINT
"▮▮":GOTO930
```

```
970  IFL=1THENHT$="0"+HT$
980  PRINT
990  INPUT"MINUTI    *■■■"; MT$
1000  A=VAL(MT$):L=LEN(MT$)
1010  IFA=0ANDMT$<>"0"ANDMT$<>"00"THENPRI
NT"TT":GOTO990
1020  IFL>2ORA<0ORA>59ORA<>INT(A)THENPRIN
T"TT":GOTO990
1030  IFL=1THENMT$="0"+MT$
1040  PRINT
1050  INPUT"SECONDI    *■■■";HS$
1060  A=VAL(HS$):L=LEN(HS$)
1070  IFA=0ANDHS$<>"0"ANDHS$<>"00"THENPRI
NT"TT":GOTO1050
1080  IFL>2ORA<0ORA>59ORA<>INT(A)THENPRIN
T"TT":GOTO1050
1090  IFL=1THENHS$="0"+HS$
1100  TI$=HT$+MT$+HS$
2000  PRINT"■■■■■■■■■■■■■■■■■■■"
2010  PRINTCL$"■":PRINTCL$"■":PRINTCL$
2020  RETURN
3000  GOSUB2000
3010  PRINT"■■■■■■■■■■■■■■■■■■■"
3020  INPUT"DATA                  *■■■";DA$
3030  D=VAL(DA$):L=LEN(DA$)
3040  IFL>2ORD<0ORD>LEN(TN$)/2ORD<>INT(D)
THENPRINT"TT":GOTO3020
3050  IFD=0THEN3130
3060  A1=PEEK(A(D)):A2=PEEK(A(D)+1)
3070  IFD=0THEN3140
3080  IFA1=32THEN3110
3090  A1=A1+128:IFA1>255THENA1=A1-256
3100  POKEA(D),A1
3110  A2=A2+128:IFA2>255THENA2=A2-256
3120  POKEA(D)+1,A2
3130  GOSUB2000
3140  RETURN
4000  GOSUB3060
4010  D=D+1:IFD>LEN(TN$)/2THEN4100
4020  GOSUB3060
4030  IFTI$="000000"THEN4030
4040  GOTO725
4100  IFTY=9999THEND=0:GOTO4020
4110  D=1:TM=TM+1:IFTM>12THENTM=1:TY=TY+1
4120  GOTO305
```

........................Cronometro

Per quanto sia apprezzabile il valore dell'espressione TI$, essa e' chiaramente inadeguata quando ci e' indispensabile misurare intervalli minori di un secondo; per questo ci viene in soccorso l'orologio interno del 64, preciso fino ad un sessantesimo di secondo. Quindi per avere unefficiente cronometro, dovremo per questa volta ignorare TI$ per guardare direttamente all'orologio interno al computer.

L'orologio e' locato negli indirizzi tra il 160 e il 162 nella memoria del 64. I contenuti dell'indirizzo 162 vengono incrementati ogni sessantesimo di secondo, e mentre si commutano da 255 (il massimo valore di un singolo byte) a 0, quelli della locazione 161 vengono incrementati di 1. Allo stesso modo, quando i contenuti dell'indirizzo 161 cambiano da 255 a 0, i contenuti in 160 si incrementano di uno. L'espressione:

```
TI$ = "000000"
```

che resetta tutti i tre valori a zero, e' equivalente a

```
POKE 160,0: POKE 161,0: POKE 162,0
```

Quindi l'espressione:

```
PEEK(162)+256*PEEK(161)+65536*PEEK(160)
```

ci calcolera' il tempo trascorso con la precisione di 1/60 di secondo. Fortunatamente tutto cio' e' accademico, poiche' il Commodore 64 possiede un'altra variabile riservata, TI, che indica il valore della espressione sopra indicata. Quello che e' necessario fare e' quindi fare in modo che il valore TI venga diviso in 1/60 di secondo per sapere precisamente il tempo trascorso.

Il solo problema per giungere a questo grado di accuratezza e' che il dato non fa in tempo ad essere visualizzato sullo schermo. Comunque, in questo programma, ho risolto il problema facendo comparire le frazioni di secondo solo dopo che il cronometro viene fermato. Invece TI$ mostra il valore (in secondi) che piu' si avvicina all'effettivo tempo registrato (la linea 100 ferma l'orologio, mentre il tempo esatto viene calcolato nelle linee dalla 200 alla 230.

Usate i tasti F1 per la partenza dell'orologio, F3 per fermarlo e mostrare l'esatto tempo trascorso, F5 per resettare il tutto (e ricominciare). Come un qualsiasi cronometro, e' possibile fermare il tempo e ripartire senza cancellarlo, per prendere anche i tempi "intermedi".

```
5 REM ***PRG.# 13-CRONOMETRO ***
10 POKE53280,1:POKE53281,1:PRINTCHR$(147
):CHR$(144):CHR$(14)
20 FORJ=1TO10:PRINT:NEXT:FORJ=1TO40:B$=B
$+CHR$(32):NEXT
30 PRINT"F1 = PARTENZA/RIPARTENZA"
40 PRINT:PRINT"F3 = STOP/TEMPO TRASCORSO
"
50 PRINT:PRINT"F5 = CANCELLA"
60 F=1
70 TI$="000000"
80 PRINTCHR$(19);TI$
90 GETI$:G=F
100 IFI$=CHR$(134)THENT=TI:F=3
110 IFI$=CHR$(133)THENF=2
120 IFI$=CHR$(135)THENTI$="000000":F=G
130 ONFGOTO70,80,200
200 H(1)=INT(T/60↑3):T=T-H(1)*60↑3
210 H(2)=INT(T/60↑2):T=T-H(2)*60↑2
220 H(3)=INT(T/60):T=T-H(3)*60
230 H(4)=INT(T*100/60)
240 PRINT:PRINTCHR$(28);"ORE   ","MINUTI
","SECONDI","CENTINAIA"
250 FORJ=1TO4:H$(J)=MID$(STR$(H(J)),2):I
FLEN(H$(J))<>2THENH$(J)="0"+H$(J)
260 PRINTCHR$(30);H$(J),
270 NEXT
280 PRINTCHR$(144)
290 GETI$:IFI$<>CHR$(133)ANDI$<>CHR$(135
)THEN290
300 PRINTCHR$(19)
310 PRINT:PRINTB$:PRINTB$
320 IFI$=CHR$(133)THENF=2
330 IFI$=CHR$(135)THENF=1
340 GOTO130
```

Archiviazione

....................Cardbox

....................Cardbox

Cardbox e' un sistema altamente sofisticato di gestione di files. Esso lavora ugualmente bene sia su disco che su cassetta, e dato che le diverse gestioni di files sono tra di loro compatibili, se per il momento possedete solo il registratore sarete sempre in tempo a trasferire i vostri dati su disco quando vi potrete permettere il sistema a dischi.

Se avete una stampante collegata al vostro Commodore 64, vi sara' possibile riprodurre su carta cio' che verra' visualizzato sullo schermo, o una lista, o, se state usando Cardbox come un'agenda di indirizzi e nomi, stamparli su delle etichette. Possono essere visualizzate schede singole o l'intera raccolta di file (in modo continuativo), oppure schede selezionate (sia singolarmente che in modo continuato).

Le schede possono essere aggiunte o cancellate, e cosi' pure i dati in esse contenute possono essere modificati. E' presente un potente sistema di ricerca che permette di ricercare i dati voluti molto velocemente; in alternativa si possono vedere le intere schede ad una ad una. Questa versione del programma non contiene una routine di ordinamento dati (SORT), ma dato che in questo stesso libro piu'avanti ve ne presentero' una sara' vostro compito, per rendere piu' completo il programma, aggiungerla.

Questo e' un vero e proprio programma da lavoro, gestiole, che e' possibile usare regolarmente per tenere un registro dei clienti e dei fornitori. Comunque potrete modificarlo a vostro piacimento e per soddisfare le vostre personali esigenze; proprio per questo vi descrivero' le caratteristiche generali delle schede, il modo in cui potrete utilizzare il mio programma e il modo migliore per modificarlo, per plasmarlo ai vostri scopi.

......................Caratteristiche generali

Ogni scheda di un file contiene nove campi (linee di informazione). I primi sei sono i campi normali che possono contenere fino a 26 caratteri ognuno. Gli altri tre sono dei campi-bandiera; cio' significa che quando verra' visualizzata la scheda, bastera' indicare con la semplice pressione di un tasto se si desidera o meno un campo gia' pre-definito (vi spieghero' meglio in seguito il funzionamento di questo semplice espediente).

Ogni file puo' contenere fino a 250 campi, ma voi potete naturalmente usare molti files (i possessori del sistema a dischi saranno indubbiamente avvantaggiati).

I campi di ogni scheda possono essere utilizzati dall'operatore per raccogliere i files di dati nel seguente modo:

Campo 1 : Nome
Campo 2 : Indirizzo (via)
Campo 3 : Indirizzo (citta')
Campo 4 : Indirizzo (provincia)
Campo 5 : Indirizzo (stato), con "Italia" come risposta suggerita dal computer
Campo 6 : Note (per ricordare qualche caratteristica importante; viene suggerito un "meno" se non si hanno commenti)
I campi-bandiera sono definiti come segue:

Campo 7 : Vendita (si/no). Usandolo siamo in grado di catalogare un cliente ed un potenziale cliente nello stesso file. Al momento della creazione di una nuova scheda, per rispondere alla domanda bastera' premere un tasto, e nella successiva visualizzazione sulla scheda comparira' la scritta ("cliente" o "nessuna vendita") in campo inverso. Da notare che le due opzioni si escludono a vicenda (cioe'o si e' clienti o non lo si e').
Campo 8 : Estinto (si/no). Un'opzione molto usata, specialmente in alternativa alla totale cancellazione della scheda dal file. Generalmente sta ad indicare un cessato rapporto di fornitura/clientela con l'intestatario della scheda. Come prima, un singolo tasto ci permette di stampare sulla scheda interessata la parola "estinto" o "corrente" (naturalmente in campo inverso).
Campo 9 : Visitato (si/no). Indica se il cliente e' stato o meno contattato da uno dei rappresentanti della nostra ditta o se esiste solo un rapporto di semplice corrispondenza.

SOLO i campi dall'1 al 5 vengono eventualmente stampati su carta (o etichetta).

Se intendete usare Cardbox semplicemente come un'agenda ddi indirizzi, penserete che siano sufficienti i primi 5 campi. Per altri scopi, per es. al fine di tenere un inventario di una vostra collezione, dovete prima di tutto cambiare le linee di testo dalla 100 alla 120; e' da notare che, poiche' la memoria e' la cosa che piu' conta in questo programma, l'elemento 0 dell'array viene usato in riferimento al campo 1, e cosi' via. Il campo 6 (elem. array 5, linea 125) e' da usarsi per dei commenti in generale, ma se volete potete dargli un argomento ben preciso (ad esempio "Numero di telefono").

Alcuni campi tra quelli dall'1 al 6 possono portare una risposta pre-determinata, in modo tale da evitare inutili e noiose ripetizioni di battiture e per scongiurare il pericolo di errore sulla scheda finale. Come gia' detto prima, i campi 5 e 6 (linee 640 e 650) usano gia' questa particolare tecnica di funzionamento. Potete cambiare o cancellare quelle linee, ma ricordatevi sempre nella manipolazione degli array che il discriminante e' sfasato di UN valore rispetto a quanto compare sul video come numero di campo.

Il meccanismo per immettere una risposta pre-determinata e' il seguente:

PRINT" [3 cursori-destra]risposta[lo stesso num. dei caratteri formanti la risposta+2]"

Questo permettera' al cursore di posizionarsi sulla primalettera della risposta, in modo tale che se il tasto RETURN viene premuto, il testo immesso e' quello. Altrimenti bastera' scrivere sopra al testo con la parola desiderata (se la vostra e' piu' corta, usate lo spazio per cancellare cio' che rimane della vecchia scritta).

I campi-bandiera (dal 7 al 9) possono essere definiti a vostro piacere alle linee 130-140. Alcuni esempi di

condizioni tra loro esclusive potrebbero essere "Mandato auguri Natale/Non mandato", "Rapporto affari/rapp. amichevole", ecc. Potreste cosi' scoprire a che avete mandato gli auguri cercando quel semplice campo.
Ricordatevi, quando definite le risposte, che dovrete immettere nel programma sia le domande per la creazione di una nuova scheda (F$) che le due risposte alternative da stampare sulla scheda definitiva (G$). Gli elementi degli array dei campi-bandiera sono SETTE IN MENO rispetto ai loro numeri di campo.

I numeri dei campi usati nel momento della stampa dipendera' dalla natura degli stessi campi. Per stampare piu' (o meno) dei primi cinque campi, cambiate in modo appropriato la subroutine alla linea 4000.

.........................Come usare Cardbox

Quando farete girare Cardbox, il menu' di apertura comparira' sullo schermo (guardate le linee 520-540). Dato che non avrete nessun dato in memoria, saranno operative solo le opzioni 1 (aggiungere schede) e 4 (caricare dati).Premete 1 e cominciate a creare una scheda, scheda che comparira' completa sullo schermo insieme ad un piccolo menu'(linea 8110). Provate con tutte le opzioni e guardate come lavorano (se non avete la stampante in funzione, non premete P(rint) o provocherete un "DEVICE NOT PRESENT ERROR"). Le opzioni N(ext) o E(scape) vi riporteranno al menu' iniziale, dal quale saranno disponibili tutte le funzioni.

Le opzioni 2 (display) e 3 (search) vi daranno l'opportunita' di stampare tutti i dati in memoria. Declinate cio' se non possedete una stampante o se non vi interessa la cosa. L'opzione 3 vi dara' l'elenco dei campi e vi chiedera' su quale campo vogliate ricercare il file. Se voi selezionate un campo da 1 'a 6 vi sara' chiesto di digitare il nome, o una parte di esso, del campo desiderato: per esempio, alla ricerca del campo "Italia", dopo aver premuto l'opzione 5 bastera' scrivere Ita o It

per avere l'elenco dei file presenti che hanno un nome simile. I campi da 7 a 9 possono essere cercati in modo specifico da un singolo tasto, che sceglie tra due condizioni estreme.

Le schede che soddisferanno il campo della vostra ricerca (tutte le schede, opzione 2) saranno visualizzate una alla volta, con ai piedi un piccolo menu' contenente comandi quali N(ext) che vi mostrera' la scheda successiva, e E(scape) che vi riproporra' la pagina iniziale.

........................Opzioni LOAD/SAVE

Il Cardbox qui listato presume che voi usiate il sistemaa dischi. Ogni file va registrato o caricato tramite il menu' iniziale, e deve possedere un nome unico. Il file viene quindi salvato sequenzialmente su disco, ed e' completamente caricato in memoria. Cio' limita l'estensione del file, ma una volta caricati i dati la loro gestione risulta essere molto veloce. Una caratteristica di questo modo di procedere e' quella che tutto viene fatto all'interno della memoria del computer; ricordatevi sempre, quando modificate un file, di ri-salvarlo su supporto magnetico.

.......................Per il sistema a cassette

Alcune linee del programma vanno cambiate per permettere a chi dispone del solo registratore di poter lavorare grazie al nostro programma:

```
9015 OPEN 9,1,1,k$
9080 CLOSE 9
9570 OPEN 9,1,0,k$
9650 CLOSE 9
```

La parola "disco" va sostituita con "nastro" alle linee 9005 e 9550. La subroutine alla linea 10000 (che legge gli eventuali errori di canale su disco) non e' richiesta e puo' essere eliminata se non pensate in un prossimo futuro di fare uso di un floppy disc. Se pensate di poterlo fare,preparatevi un programma che permetta di caricare dati da nastro e riversarli su disco. Vi bastera' cambiare le linee 9015 e 9080 per tornare al programma principale.

Poiche' Cardbox opera su files memorizzati nella memoria del computer, le operazioni tramite i sistemi a disco o a cassette sono identiche. Portera' naturalmente via piu' tempo l'uso del nastro (purtroppo il sistema e' da considerarsi peggiore principalmente per questo motivo), ma oltre a cio' l'unico inconveniente consiste nel fatto che vi sara' piu' utile usare UN nastro per ogni file (se le 250 schede basteranno ai vostri propositi, non avrete problemidi questo tipo). Le cassette dedicate al computer potrebbero risultare troppo brevi per l'uso che ne dovrete fare percio' vi consiglio di munirvi di cassette con una autonomia abbastanza lunga.

```
3 REM *** PRG.#14-ARCHIVIAZIONE ***
5 FORJ=0TO38:B$=B$+CHR$(32):NEXT
10 V$="]"+LEFT$(B$,15)+CHR$(5)+"V&H CARD
BOX"+CHR$(144)
15 M=250
20 C1$=CHR$(5)+CHR$(147)+LEFT$(B$,26)
25 C1$=C1$+CHR$(117):FORJ=1TO8:C1$=C1$+C
HR$(96):NEXT:C1$=C1$+CHR$(105)
30 C2$=LEFT$(B$,4)
35 C2$=C2$+CHR$(117):FORJ=1TO21:C2$=C2$+
CHR$(96):NEXT:C2$=C2$+CHR$(107)
40 C2$=C2$+LEFT$(B$,8)+CHR$(98)
45 C3$=LEFT$(B$,4)
50 C3$=C3$+CHR$(98)+LEFT$(B$,30)+CHR$(98
)
55 C4$=LEFT$(B$,4)
60 C4$=C4$+CHR$(106):FORJ=1TO30:C4$=C4$+
CHR$(96):NEXT
65 C4$=C4$+CHR$(107)+CHR$(144)
70 POKE53280,2
80 POKE53281,10
90 DIMA$(M,5),F(M,2)
100 D$(0)="NOME■■■■■■■■■"
105 D$(1)="VIA■■■■■■■■■■"
110 D$(2)="CITTA'■■■■■■"
115 D$(3)="PROVINCIA■■■"
120 D$(4)="STATO■■■■■■■■"
125 D$(5)="NOTE■■■■■■■■■"
130 F$(0)="VENDITA (S/N)  ":G$(0)="CLIENT
E":G$(1)="■NON IN VENDITA■"
```

```
135  F$(1)="ESTINTO (S/N) ":G$(2)="ESTIN
TO":G$(3)="CORRENTE"
140  F$(2)="VISITATO (S/N) ":G$(4)="CHIAM
ATA PERSONALE":G$(5)="NON VISTO"
150  N=0
500  P=0:PRINTV$
510  PRINT"NUMERO DI SCHEDE="N"
520  PRINT"1:AGGIUNGI NUOVE SCHEDE":PRI
NT"2:ELENCA TUTTE LE SCHEDE"
530  PRINT"3:RICERCA TRAMITE SCHEDE"
540  PRINT"4:CARICA FILE":PRINT"5:SAL
VA FILE"
550  GETI$:IFI$<"1"ORI$>"5"THEN550
560  I=VAL(I$):IFI>1ANDI<>4ANDN=0THEN550
570  ONIGOTO600,2000,1000,9500,9000
600  IFN=MTHEN PRINT"SCHEDE FINITE":FOR
J=1TO2000:NEXT:GOTO500
610  N=N+1:PRINT"SCHEDE #"N""
620  FORJ=0TO5
630  PRINTD$(J);:IFJ<4THENPRINT"▶▶▶■■■";
640  IFJ=4THENPRINT"▶▶▶ITALIA■■■■■■■■■";
650  IFJ=5THENPRINT"▶▶▶-■■■";
660  INPUTT$
670  IFT$="*"THENPRINT"":GOTO630
680  IFLEN(T$)>26THENT$=LEFT$(T$,26)
690  A$(N,J)=T$:PRINT
700  NEXT
710  FORJ=0TO2
720  PRINTF$(J);
730  GETI$:IFI$<>"S"ANDI$<>"N"THEN730
740  PRINTI$:F(N,J)=ASC(I$)
750  NEXT
760  C=N
770  GOSUB8000
780  GOTO500
1000  GOSUB3000
1010  PRINTV$:PRINT"I CAMPI SONO:"
1020  FOR J=0TO5:PRINTJ+1;D$(J):NEXT
1030  FOR J=6TO8:PRINTJ+1;F$(J-6):NEXT
1040  PRINT"QUALE CAMPO CERCHI? (0=PER
FINIRE)"
1050  GETI$:IFI$<"0"ORI$>"9"THEN1050
1060  V=VAL(I$)-1:IFV=-1THEN500
1070  IFV>5THEN1200
1080  PRINT"QUALE SPECIFICA NEL CAMPO"V+
1""
1090  INPUT"▶▶▶■■■";Q$:IFQ$="*"THENPRINT
"":GOTO1090
1100  F=0:C=0
1105  C=C+1
1110  S$=A$(C,V):S=LEN(S$)-LEN(Q$)+1
1120  FORX=1TOS
1130  IFMID$(S$,X,LEN(Q$))=Q$THENGOSUB800
0:X=S:F=1
1140  NEXT:IFC<NTHEN1105
1150  IFF=0THENPRINTTAB(13)"NESSUN CAMPO
 TROVATO":FOR J=1TO2000:NEXT:GOTO1010
1160  GOTO1500
1200  PRINTV$
1210  PRINT"NEL CAMPO"V+1":"CERCO"
1220  PRINT"'S'="G$((V-6)*2)
1230  PRINT"'N'="G$((V-6)*2+1)
1240  PRINT"RICERCA   'S' O 'N'?"
1250  GETI$:IFI$<>"S"ANDI$<>"N"THEN1250
1260  S=ASC(I$)
1270  F=0:FORC=1TON
1280  IFF(C,V-6)=STHENGOSUB8000:F=1
```

```
1290  NEXT
1300  GOTO1150
1500  PRINTV$"■■■":PRINTTAB(9)"NON PIU' S
CHEDE DA CERCARE"
1520  FORJ=1TO2000:NEXT:GOTO500
2000  GOSUB3000
2040  C=0
2050  C=C+1:GOSUB8000
2060  IFC<NTHEN2050
2070  GOTO500
3000  PRINT"■■VUOI CONTINUARE A VISUALIZZ
ARE? (S/N)"
3020  P=0:GETI$:IFI$="S"THENP=1:RETURN
3030  IFI$<>"N"THEN3020
3040  RETURN
4000  OPEN4,4:CMD4
4010  FORJ=0TO4:PRINTA$(C,J):NEXT:PRINT#4
:CLOSE4
4020  RETURN
8000  PRINTC1$:PRINTC2$
8010  FORJ=1TO16:PRINTC3$:NEXT
8020  PRINTC4$
8030  PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
■■■■■■■■■■■■■■■■■■■■■"C"■"
8040  FORJ=0TO5:PRINTJ+1":■■■"A$(C,J)"■":
NEXT
8050  FOR J=6TO8:PRINTJ+1":■■■■■■■■■■■■"G$
((J-6)*2-(F(C,J-6)=78)):NEXT
8100  PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■"
8110  PRINT" ■C■AMBIO: ■E■LIMINA: ■S■TAMP
A: ■P■OST : ■U■SCITA■"
8120  GETI$:IFI$="C"THEN8200
8130  IFI$="E"THEN8300
8140  IFI$="S"THENGOSUB4000:GOTO8120
8150  IFI$="P"THENRETURN
8160  IFI$="U"THENC=N:RETURN
8165  IFP=1THENGOSUB4000:RETURN
8170  GOTO8120
8200  PRINT"QUALE CAMPO CAMBI? (0=STOP)"
8210  GETI$:IFI$<"0"ORI$>"9"THEN8210
8220  IFI$="0"THENPRINT"■"B$:GOTO8100
8230  I=VAL(I$)-1:IFI>5THENF(C,I-6)=89+11
*(F(C,I-6)=89):GOTO8000
8240  INPUT"NUOVI DATI■■■■■■■■■";T$
8250  IFT$="*"THENPRINT"■■":GOTO8240
8260  IFLEN(T$)>26THENT$=LEFT$(T$,26)
8280  A$(C,I)=T$
8290  GOTO8000
8300  PRINT"SEI SICURO? (S/N) : 'S' CANCE
LLA SCHEDA": RUSCITAQ"
8310  GETI$:IFI$="N"THENPRINT"■"B$:GOTO81
00
8320  IFI$<>"S"THEN8310
8340  FORJ=1115TO1736STEP41:POKEJ,230:NEX
T
8350  FORJ=1130TO1736STEP39:POKEJ,230:NEX
T
8360  PRINT"■■■"B$"■"B$"■":PRINTTAB(15)"■
CANCELLO■"
8365  N=N-1:IFC=NTHEN8500
8370  FORJ=CTON:FORK=0TO5
8375  A$(J,K)=A$(J+1,K):NEXT:FORK=0TO2
8380  F(J,K)=F(J+1,K):NEXTK,J
8390  C=C-1
8500  RETURN
9000  PRINTV$:INPUT"■NOME FILE■■■■■■■";K$:
IFK$="*"THENK$="FILE"
9005  PRINT"■METTI DISCO: PREMI 'SPACE' Q
```

```
UANDO OK"
9010 GETI$:IFI$<>CHR$(32)THEN9010
9015 OPEN9,8,9,"@0:"+K$+"S,W"
9020 PRINT"SALVO "K$
9025 PRINT#9,N
9030 FORJ=1TON:FORK=0TO5
9040 PRINT#9,A$(J,K)
9050 NEXT:FORK=0TO2
9060 PRINT#9,F(J,K)
9070 NEXTK,J
9080 CLOSE9:GOSUB10000
9090 GOTO500
9500 IFN=0THEN9540
9510 PRINT"■SEI SICURO? (S/N): ´S´ CANC.
I FILE"
9520 GETI$:IFI$="N"THEN500
9530 IFI$<>"S"THEN9520
9540 PRINTV$:INPUT"■NOME FILE■■■■■■■";K$:
IFK$="*"THENK$=""
9550 PRINT"■METTI DISCO: PREMI ´SPACE´ Q
UANDO OK "
9560 GETI$:IFI$<>CHR$(32)THEN9560
9570 OPEN9,8,9,"0:"+K$+"S,R"
9580 INPUT#9,N
9590 PRINT"CARICO "K$:PRINT"■"N"CAMPI"
9600 FORJ=1TON:FORK=0TO5
9610 INPUT#9,A$(J,K)
9620 NEXT:FORK=0TO2
9630 INPUT#9,F(J,K)
9640 NEXTK,J
9650 CLOSE9:GOSUB10000
9660 GOTO500
10000 OPEN15,8,15
10010 INPUT#15,E1,E2$,E3,E4
10020 IFE1=0THEN10070
10030 PRINT"□DISC ERROR:■"
10040 PRINTE1:PRINTE2$:PRINTE3:PRINTE4
10050 PRINT:PRINT"PREMI UN TASTO PER CON
TINUARE"
10060 GETI$:IFI$=""THEN10060
10070 CLOSE15:RETURN
```

Utili routines

- Subroutines
- Print @
- Col @
- Auto-line
- Auto-delete
- Auto-renumber
- Sort
- Format

..........................Subroutines

Questo programma e' in effetti una collezione di subroutines, alcune delle quali troverete (almeno credo) molto utili. Comunque, la ragione ha come scopo l'illustrazione di un'idea che, indipendentemente da cio' che pensiate su queste particolari routines, troverete molto utile.

L'idea consiste nel fatto che probabilmente ognuno sarebbe in grado di costruirsi una biblioteca di subroutines utili, e che ho pensato che sarebbe stato meglio avere a disposizione in un unico programma una serie di utility pronte all'uso.
Eviterete cosi' di caricare ogni volta la particolare routine che vi interessa e vi troverete a portata di mano tutto quanto e' necessario ad una buona programmazione.

Scoprirete che, usando questa tecnica, i numeri di linea che generalmente voi assegnavate alle subroutine restano nella vostra mente, e che vi bastera' battere "GOSUB 5" per richiedere di premere un tasto, cosi' facilmente come il resto dei comandi Commodore. Creare una raccolta di subroutines e' il modo piu' semplice per arricchire il BASIC del 64.

......................Note del programma

Lo scopo delle varie subroutines viene spiegato nelle linee 80-87, e dovreste prenderla anche voi come abitudine quella di aggiungere delle istruzioni REM per ricordare dove si trovi il comando che a voi serve. Questo vi aiutera' anche al termine della programmazione, quando dovrete cancellare tutte le linee che non sono piu' utili al programma, o che non avete usato.

Le linee dalla 100 alla 260 sono una semplice dimostrazione di quali siano le caratteristiche delle varie subroutines in azione, in modo tale che siano necessarie solo poche altre parole per impararne l'uso. La subroutine 30 permette l'uso di un "carattere programmato"

su una matrice di punti di una stampante che ammetta questa operazione (molte stampanti Commodore possono). Alcune stampanti ad es. non posseggono il segno "/", e questo handicap puo'essere risolto grazie a queste poche linee di programma.

```
0 REM *** PRG.#15- SUBROUTINE ***
1 GOTO100
5 PRINTTAB(4)"PREMI UN TASTO PER CONTINU
ARE"
6 Q$="":GETQ$:IFQ$=""THEN6
7 RETURN
10 B$=CHR$(145):FORJ=1TO39:B$=B$+CHR$(32
):NEXT:B$=B$+CHR$(145)
11 PRINTB$:RETURN
15 FORK=1TOLEN(C$):IFMID$(C$,K,1)="."THE
N17
16 NEXTK:C$=C$+".00":RETURN
17 L=LEN(C$)-K
18 IFL=0THENC$=C$+"00":RETURN
19 IFL=1THENC$=C$+"0":RETURN
20 IFL>2THENC$=LEFT$(C$,LEN(C$)+2-L):RET
URN
25 BR=14:BG=6:PC=154
26 POKE53280,BR:POKE53281,BG:PRINTCHR$(P
C);
27 RETURN
30 OPEN9,4,5:FORK=1TO6:READA:PS$=PS$+CHR
$(A):NEXT:PRINT#9,PS$:CLOSE9:RETURN
31 DATA0,9,63,73,65,33
80 REM GOSUB5 = GET ANY KEY (WITH MESSAG
E)
81 REM GOSUB6 = GET ANY KEY (WITHOUT MES
SAGE)
82 REM GOSUB10 = DELETE CURRENT SCREEN L
INE
83 REM GOSUB11 = AS 10 (SUBSEQUENTLY)
84 REM GOSUB15 = CASH FORMATTING OF C$
85 REM GOSUB25 = RESET COLOURS TO ORIGIN
AL
86 REM GOSUB26 = SET COLOURS AS SPECIFIE
D BY BR, BG AND PC
87 REM GOSUB30 = POUND SIGN FOR PROGRAMM
ABLE PRINTER : CHR$(254)
100 PRINTCHR$(147):REM DEMOS
110 BR=0:BG=0:PC=5:GOSUB26
120 PRINT"CIAO"
130 FORJ=1TO300:NEXT
140 GOSUB10
150 GOSUB5
160 GOSUB11
170 PRINT"ARRIVEDERCI"
180 INPUT"DIGITA DEI VALORI (PER LA FORM
ATTAZIONE)";C$
190 GOSUB15:PRINT"FORMATTATO:"C$
200 PRINT"POSSIEDI UNA STAMPANTE PROGRAM
MABILE?"
210 GOSUB6
230 IFQ$="N"THEN260
240 IFQ$<>"S"THEN210
250 GOSUB30:OPEN4,4:PRINT#4,CHR$(254)C$:
CLOSE4
260 GOSUB25
```

...............................Print @

Se avete usato altri micro-personal prima di acquistare il Commodore 64, probabilmente avrete sentito la mancanza dell'istruzione PRINT @ (o, su altri computer, PRINT AT). Questa e' una semplice istruzione BASIC che posiziona il cursore e la successiva stampa-dati in un punto ben preciso dello schermo. Per esempio:

```
PRINT @ 0;"CIAO"
```

stamperebbe la parola "CIAO" alla posizione HOME. Ma non sul Commodore 64!

Per posizionare il cursore usando il BASIC Commodore e' necessario usare i caratteri determinati dai controlli-cursore (tutti quelli irritanti caratteri inversi di cui sono pieni molti listati) o usare una lunga ed ingombrante serie di CHR$(17), ecc. Questa subroutine risolve tutti questi problemi simulando la funzione PRINT @.

(Naturalmente non potrete usare questa funzione chiedendo direttamente all'interprete BASIC di eseguirla; dovrete invece scrivere "P=x: GOSUB 1000", dove x e' la posizione nella quale voi volete scrivere)

ad esempio : P=0: GOSUB 1000: PRINT"CIAO"

avra' il medesimo effetto della suddetta funzione PRINT@.

...........................Note del programma

La linea 1010 chiama una seconda subroutine che inizializza le due stringhe di 39 cursori a destra (A$) e di 29 cursori in basso (D$). Una volta che queste due stringhe sono state create, questa parte del programma non verra' piu' eseguita.

La linea 1030 lavora in modo tale che il BASIC Commodore riconosca LEFT$(D$,D) e LEFT$(A$,A) come espressioni legittime, anche quando A o D risultino uguali a 0.

L'espressione LEFT(D$,0) dovrebbe apparire come un'istruzione impossibile da eseguire, ma grazie a questo espediente cio' e' possibile con la creazione di una stringa nulla, che non produce alcun effetto sullo schermo.

Le linee 10-30 non fanno parte della routine, e dimostrano semplicemente come si possano stampare dei Cuori a piacimento sullo schermo. Notate che i cuori sono stampati vicini, a dispetto del fatto che non ci sia una semi-colonna al termine della linea 20.

Un buon suggerimento per voi: alcuni dialetti BASIC permettono che l'espressione PRINT @ sia seguita da due numeri, rappresentanti le coordinate x e y della posizione di schermo richiesta (molto utile per i programmatori che hanno difficolta' a dividere per 40). Cercate di modificare la routine in modo tale che provveda in tal senso.

```
5 REM *** PRG.#16- PRINT @ ***
10 FORP=0TO959
20 GOSUB1000:PRINTCHR$(115)
30 NEXT:END
1000 REM P=...:GOSUB1000 = PRINT @ P
1010 IFA$=""THENGOSUB1050
1020 D=INT(P/40):A=P-40*D
1030 PRINTCHR$(19)+LEFT$(D$,D)+LEFT$(A$,
A);
1040 RETURN
1050 FORJ=1TO39:A$=A$+CHR$(29):NEXT
1060 D$="":FORJ=1TO24:D$=D$+CHR$(17):NEX
T
1070 RETURN
```

..................................Col @

E' abbastanza semplice rammentere i codici-colore (i numeri 0-15) che rappresentano le possibili combinazioni effettuabili coi colori sul Commodore 64. Non e' semplice

ricordare il corrispondente codice ASCII (riguardatevi la tabella a pag. 12 se non vi rammentate nulla). Di conseguenza, cambiando il colore di stampa diventa piu' difficile che cambiare il colore del bordo. Voi potete, come ho gia' detto nel primo capitolo, usare la POKE 646 per cambiare questi colori, ma c'e' anche la locazione POKE da ricordare.

Sareste in grado di scrivere

PRINT COL$(6)

per cambiare il colore di stampa in blu? Usate questa subroutine una volta e lo potrete fare.

```
10 REM *** PRG.#17- COL@ ***
2000 DIMCOL$(15)
2010 FORJ=0TO15
2020 READC
2030 COL$(J)=CHR$(C)
2040 NEXT
2050 RETURN
2060 DATA144,5,28,159,156,30,31,158,129,
149,150,151,152,153,154,155
```

....Auto-line

Questo e' un programma di utility che vi da' accesso ad una serie di routines riguardanti le linee di programma o una porzione comune delle linee stesse.

Per usare "Auto-line", caricate il programma PRIMA di iniziare la digitazione di un programma. Quindi, arrivati ad una serie di linee contenenti una massiccia dose (ad es.) di istruzioni DATA o PRINT o ogni altra espressione molto ripetuta, battete "RUN 10000".

Vi sara' chiesto di immettere il numero di linea di partenza, l'incremento da dare e l'istruzione BASIC con la quale ogni linea deve cominciare (poiche' DATA e PRINT" sono i due esempi piu' ovvi, potete selezionarli con la semplice pressione di un tasto). Il numero di linea e la parola scelta appariranno sullo schermo, con il cursore

posizionato in modo tale da permettere l'entrata del successivo testo riguardante la linea di programma. Ad ogni pressione del tasto RETURN la linea sara' memorizzata e comparira' la linea successiva con la medesima prima istruzione.

Per uscire dal programma premete RUN/STOP e il vostro programma, insieme alle linee che avete appena digitato, verra' listato sullo schermo.

............................Note del programma

Questo programma sfrutta una delle caratteristiche del Commodore 64. Ogni volta che premete un tasto, il codice ASCII di quel tasto viene registrato nella memoria del computer, fino al momento in cui verra' richiamato. Possono essere memorizzati fino a dieci tasti in questo modo. In Editing (quando cioe' non gira alcun programma) avete un rapporto diretto tra la tastiera e lo schermo, ed un carattere viene visualizzato appena dopo battuto. Comunque dovete sapere che quando gira un programma e' spesso possibile premere diversi tasti prima che vi venga richiesto dal computer, che ne terra' conto se intendeva proporvi un'istruzione INPUT o GET.

Il buffer di tastiera e' locato agli indirizzi 631-640, e un'altra locazione, l'indirizzo 198, indica il numero dei caratteri memorizzati nel buffer. Normalmente i caratteri entrano nel buffer grazie al solito metodo (premendo un tasto) ma e' anche possibile, tramite POKE, immetterli tramite un programma.

La linea 10210 simula (con POKE) tre caratteri RETURN (il carattere ASCII e' 13) nel buffer di tastiera e il numero 3 nella locazione 198 segnala che tre valori sono memorizzati nel buffer. A questo punto, se non ci fossero istruzioni INPUT o GET, i comandi verrebbero effettuati senza particolari effetti. Ma date un'occhiata a quello che accade nel frattempo:

La linea 10100 stampa il numero di linea e la parola in BASIC precedentemente scelta. Le linee 10120-10160 vi permettono di terminare la linea di istruzione, come programmate di solito, e quando l'operatore preme RETURN (CHR$(13) alla linea 10150), le linee 10180 e 10220 stampano la seguente informazione sullo schermo:

A = [pross.num.di linea]: N = [incremento]:
A$ = [comando BASIC]

GOTO 10100

Quindi alla linea 10220 il programma finisce. Ma ironicamente l'espressione FINE e' proprio dove il programma comincia ad essere interessante. Non appena il programma e' finito, vi ritrovate in editing ed ogni comando memorizzato nel buffer di tastiera viene immediatamente visualizzato, a cominciare dai tre RETURN che avevate inserito tramite POKE precedentemente. Dopo che il cursore e' apparso in HOME, subito prima della fine del programma, l'effetto dei tre comandi RETURN e' lo stesso che si ha dopo la digitazione di tre linee di programma sullo schermo. In altre parole:

1 - Il comando BASIC che avete appena dato e' stato incorporato nel programma (primo RETURN)

2 - I valori di A, N e A$ sono memorizzati in tre locazioni (secondo RETURN). Questo passo e' necessario perche' il primo punto ha alterato il programma, causando la perdita e l'azzeramento di tutte le variabili.

3 - La linea "GOTO 10100" viene eseguita (terzo RETURN)

Cosi', non appena il programma sembra essere finito, ecco che riparte nuovamente dalla linea 10100.

Una seconda caratteristica di questo programma e' la re-definizione del tasto RUN/STOP. Premendo questo tasto normalmente si provoca uno stop al programma, ma e' possibile disabilitarlo mediante il comando

POKE 808,225

Molti programmi commerciali includono questa istruzione per evitare incidenti da parte degli operatori inesperti. Per riabilitare il tasto RUN/STOP occorre aggiungere:

POKE 808,237

Una volta che RUN/STOP e' stato disabilitato, il tasto stesso puo' essere usato per un'istruzione GET; RUN (shiftato) ha un codice ASCII di 131, e STOP (non sciftato) ha un codice ASCII di 3. La linea 10120 controlla quando questo tasto viene premuto (dopo che e' stato disabilitato alla linea 10110) e provoca il salto alla linea 10300 che lista il programma. Il ciclo di ritardo alla linea10300 e' necessario per permettere all'operatore di usare il tasto RUN/STOP prima che venga di nuovo disabilitato, altrimenti il comando LIST non verrebbe mai raggiunto.

```
10 REM *** PRG.# 18- AUTO-LINE ***
10000 PRINTCHR$(147)"INIZIO AUTO-LINE";C
HR$(17)
10010 INPUT"PRIMO NUMERO DI LINEA";A
10020 INPUT"INCREMENTO IN <STEPS> DI";N
10030 PRINTCHR$(17)"1=DATA"
10040 PRINTCHR$(17)"2=PRINT"+CHR$(34)
10050 PRINTCHR$(17)"3=ALTRI (SPECIFICA)"
CHR$(17)
10060 GETQ$:IFQ$<"1"ORQ$>"3"THEN10060
10070 Q=VAL(Q$):IFQ=1THENA$="DATA":GOTO1
0100
10080 IFQ=2THENA$="PRINT"+CHR$(34):GOTO1
0100
10090 INPUT"COMANDO";A$
10100 PRINTCHR$(147);CHR$(17);CHR$(17);A
;A$;
10110 POKE808,225
10120 GETQ$:IFQ$=CHR$(3)ORQ$=CHR$(131)TH
EN10300
10130 IFQ$=""THEN10120
10140 POKE808,237
10150 IFQ$=CHR$(13)THEN10170
10160 PRINTQ$;:GOTO10120
10170 C=0:IFRIGHT$(A$,1)=CHR$(34)THENA$=
LEFT$(A$,LEN(A$)-1):C=1
10180 PRINT:PRINT"A="A+N;":N="N;":A$="CH
R$(34);A$;CHR$(34);
10190 IFC=1THENPRINT"+CHR$(34)";
10200 PRINT
10210 POKE198,3:POKE631,13:POKE632,13:PO
KE633,13
10220 PRINTCHR$(17)CHR$(17)"GOTO 10100"C
HR$(19);:END
10300 FORJ=1TO400:NEXT:POKE808,237:LIST
```

...........................Auto-delete

Questo programma vi permettera' di cancellare diverse linee di programma consecutive. Come esempio estremo, puo' cancellare un intero programma (anche se sarebbe molto piu' facile usare l'istruzione NEW), ma viene principalmente usato in fase di editing per eliminare un'intera sezione del programma non voluta. Esso non puo' cancellare totalmente se stesso poiche' se viene cancellata la linea 10050 non viene piu' eseguita l'istruzione GOTO alla linea 10060, ed il programma si ferma. A questo punto dovrete cancellare manualmente le poche righe che rimangono.

Usate il programma nello stesso modo di Auto-line: caricatelo prima di programmare e quando vi serve la sua funzione, digitate "RUN 10000", seguito dal primo numero di linea che va cancellato, dall'incremento (se avete usato diversi incrementi, inserite il valore 1), e dall'ultimo numero di linea dove il programma si deve fermare.

...........................Note del programma

Se avete letto la spiegazione del programma precedente, allora il procedimento di questo nuovo programma vi sara' gia' chiaro in gran parte; esso non e' altro infatti che un altro spettacolare esempio di come usare il buffer di tastiera.

Questa volta l'operatore , avendo specificato i parametri iniziali, non deve assolutamente intervenire poiche' il programma si modifica da solo senza bisogno dell'intervento umano. Due linee di informazione vengono visualizzate sullo schermo:

[numero di linea da cancellare]

```
A=[linea+incremento]:B=[linea finale]:N=[incremento]
GOTO 10050
```

Il comando END alla linea 10090 rimette in editing il computer, dopo che i due caratteri RETURN sono stati fatti eseguire (linea 10080) dal buffer di tastiera, causando l'esecuzione delle due istruzioni poste sopra.

Il numero di linea visualizzato causa l'eliminazione della linea stessa (primo RETURN) mentre l'altra istruzione fa rientrare le variabili ed effettua il "GOTO 10050", facendo proseguire il programma (secondo RETURN).

```
10000 REM *** PRG. #19 - AUTO DELETE ***
10001 PRINTCHR$(147):FORJ=1TO10:PRINT:NE
XT
10010 PRINT"AUTO-DELETE";CHR$(147)
10020 INPUT"PRIMO NUMERO DI LINEA";A
10030 INPUT"INCREMENTO IN <STEPS> DI";N
10040 INPUT"ULTIMO NUMERO LINEA";B
10050 PRINTCHR$(19);CHR$(17);CHR$(17);A
10060 PRINT"A="A+N":B="B":N="N":GOTO1005
0"
10070 IFA>BTHENPRINTCHR$(147):LIST
10080 POKE198,2:POKE631,13:POKE632,13
10090 PRINTCHR$(19);:END
```

.........................Auto-renumber

Questa e' una routine che rinumera le linee di tutto il programma seguendo un passo scelto dall'operatore. Basta digitare i quattro parametri richiesti: il primo numero di linea da cambiare, il nuovo numero di pertenza, l'incremento richiesto, ed il numero dell'ultima linea da rinumerare. Quando la rinumerazione e' stata completata, l'intero programma viene listato.

Solo i numeri di linea che precedono le istruzioni verranno rinumerate. Tutti i GOTO e i GOSUB dovranno essere corretti manualmente con i nuovi valori.

........................Note del programma

La locazione nella quale il programma BASIC viene immagazzinato nella memoria del Commodore 64 puo' essere scoperta facendo riferimento agli indirizzi 43 e 44. Il programma non e' memorizzato in quelle locaziomi; ci sono semplicemente dei puntatori che rivelano le locazioni di memoria occupate, attraverso la formula PEEK(43)+256*PEEK(44)

Questa rivelera' il punto di partenza del programma BASIC; la fine, naturalmente, dipendera' dalla sua lunghezza, ma e' possibile scoprirla tramite PEEK(45)+256*PEEK(46)

Questa seconda formula, che non e' attualmente richiesta dal programma di rinumerazione, legge tramite PEEK il puntatore della variabile memorizzata, che e' sempre locata immediatamente dopo la fine di un programma BASIC.

Ogni linea di programma viene memorizzata nel seguente modo:

[2 bytes contenenti l'indirizzo della linea successiva] seguiti immediatamente da [2 bytes contenenti la linea di numero corrente] seguiti immediatamente da [il testo della linea]

Cosi', se noi chiamiamo l'indirizzo di partenza di un programma BASIC "S" (vedi linea 10050), PEEK(S+2)+256*PEEK(S+3) sara' il primo numero di linea, e PEEK(S)+256*PEEK(S+1) sara' l'indirizzo della linea successiva.

Nell'ultima linea di programma i due bytes contenenti il successivo indirizzo hanno entrambi valore 0, che e' un indirizzo impossibili per una linea di programma. Se durante il funzionamento di Auto-renumber questo punto fosse ricercato, la linea 10080 (IF S1=0...THENLIST) riconosce che la fine del programma e' stata trovata e

pone fine al programma terminando la rinumerazione. In pratica, voi sarete spiacevolmente costretti a rinumerare le linee di programma con valori superiori a quelli dello stesso Auto-renumber, perche' l'altra parte della linea 10080 (...OR L>B...) controlla che il numero di linea corrente sia piu' alto dell'ultima linea della routine.

N.B.: Auto-renumber, Auto-delete e Auto-line sono tutti listati in questo libro a partire dalla linea 10000. Voi potete, naturalmente, rinumerare due di loro per immetterli tutti e tre in memoria nello stesso momento, potendo contare cosi' su di un potente aiuto alla programmazione. Inseriteli in memoria prima di digitare i vostri programmi, ed al termine usate l'Auto-delete per cancellarli.

```
10000 REM *** PRG. #20 - AUTO RENUMBER *
**
10001 PRINTCHR$(147)"LINE RENUMBER";CHR$
(17)
10010 INPUT"PRIMO NUMERO DI LINEA (VECCH
IO)";A
10020 INPUT"PRIMO NUMERO DI LINEA (NUOVO
)";R
10030 INPUT"INCREMENTO IN <STEPS> DI";N
10040 INPUT"ULTIMO NUMERO DI LINEA (VECC
HIO)";B
10050 S=PEEK(43)+256*PEEK(44)
10060 S1=PEEK(S)+256*PEEK(S+1)
10070 L=PEEK(S+2)+256*PEEK(S+3)
10080 IFS1=0ORL>BTHENLIST
10090 IFL<ATHENS=S1:GOTO10060
10100 R1=INT(R/256):R2=R-R1*256:R=R+N
10110 POKES+2,R2:POKES+3,R1
10120 S=S1:GOTO10060
```

................................Sort

I programmi di riordinamento alfabetico o numerico sono una parte vitale nelle operazioni di gestione dati.Questo programma illustra come in modo efficace il Commodore 64 possa riordinare una serie di numeri in ordine crescente. Vi daro' le necessarie indicazioni per modificare il programma per avere inoltre:

a) Riordinamento decrescente numerico

b) Riordinamento in ordine alfabetico di caratteri-stringa

La subroutine che parte dalla linea 1000 e' tutto quanto vi e' necessario per il processo di riordinamento; tutto il resto e' una semplice cornice. Il programma vi da' l'opportunita' di trattare il riordinamento di 20 numeri casuali scelti dal computer o di numeri digitati da voi (in ogni caso, se usate piu' di venti numeri dovrete modificare la schermata che li visualizza).

......................Note del programma

Non servono parole per spiegare il funzionamento di questa routine, perche' piu' avanti potrete trovare un aiuto visivo che vi illustrera' graficamente cio' che esattamente succede.

```
5 REM *** PRG.#21- SORT ***
10 PRINTCHR$(147)
20 PRINT"SORT ROUTINE"
30 PRINT"1: DEMO"
40 PRINT"2: ENTRATA DATA"
50 GETI$:IFI$<"1"ORI$>"2"THEN50
60 ONVAL(I$)GOTO100,200
100 N=20
110 DIMS(N)
120 FORJ=1TON
130 S(J)=INT(RND(1)*1000)
140 NEXT
150 GOTO300
200 INPUT"QUANTI NUMERI ";N
210 DIMS(N)
220 FORJ=1TON
230 INPUTS(J)
240 NEXT
300 PRINTCHR$(147);"DISORDINATO",,"ORDIN
ATO"
310 FORJ=1TON:PRINTS(J):NEXT
320 A=VAL(TI$):GOSUB1000:A=VAL(TI$)-A
330 PRINTCHR$(19)
340 FORJ=1TON:PRINTTAB(20);S(J):NEXT
350 PRINTA"SECONDS":END
1000 FORJ=1TON:S=0:FORK=1TON-J
1010 T=S(K):IFT>S(K+1)THENS(K)=S(K+1):S(
K+1)=T:S=1
1020 NEXTK:IFS=0THENJ=N
1030 NEXTJ:RETURN
```

..............................Format

Questo programma serve esclusivamente ai possessori di un sistema a dischi. Mi spiace per il resto di voi che in questo paragrafo non troveranno forse nulla d'interessante.

Presumendo che non sia la prima volta che prendete in mano un floppy disc, penso che sappiate che un disco prima dell'uso va formattato (dovete essere certi della fedelta' del disco in vostro possesso, e soprattutto trattarlo con scrupolosita' per essere certi che riesca a non perdere i data che voi sopra conserverete). Formattare un disco e' molto semplice e abbisogna di un'unica istruzione; il probleme si pone quando noi per errore perdiamo il contenuto di un disco perche' non ci siamo accorti che conteneva gia' dei dati. Questo perche':

1) il comando di formattazione e' identico a quello di ri-formattazione. Quante volte ci e' capitato di perdere irreparabilmente dei dati in questo modo ?

2) ogni errore compiuto all'atto della formattazione puo' giocare qualche brutto scherzo anche diverso tempo dopo, pregiudicando la conservazione dei dati.

Un terzo problema, se avete una memoria pessima come la mia, consiste nel ricordarsi quale sia l'istruzione di formattazione senza dover andare ogni volta a controllarne l'esattezza su qualche manuale.

Questo programma, che voi potete salvare su disco, nellasperanza che vi fidiate facendolo girare con il disco nel drive, vi permettera' di formattare, ri-formattare, controllare precedentemente se in esso siano o meno contenuti dei dati, se il disco ha un'esatta intestazione, ecc.

Infine, dato che non e' una buona idea quella di dare lo stesso nome e lo stesso numero di identificazione (ID) a

piu' di un disco, il programma controlla i valori difettosi e aiuti i molto pigri ad inserire una coppia di valori (basta premere RETURN).

......................Note del programma

Il fulcro del programma e' nella linea 90, che attende tramite una GET un carattere dalla directory dal disco inserito nel drive. Se raccoglie un carattere, significa che e' presente una directory, e viene visualizzato un messaggio di allarme (linee 120-150). Altrimenti, senza la presenza di una directory, puo' proseguire il processo di formattazione (linea 110).

Questo e' un interessante capovolgimento del concetto di errore. Infatti quando, normalmente, si riceve un "OK" significa che non ci sono errori. Nel nostro caso, invece, la presenza di un "OK" ci avverte che il disco e' gia' formattato, e compare un avviso di errore. Solo se arriva un messaggio di errore il programma puo' proseguire.

```
5 REM ***PRG.#22 - FORMAT ***
10 POKE53281,1:PRINTCHR$(144)
20 N$="DISCO":INPUT"NOME DISCO";N$
30 C$="01":INPUT"CODICE D' IDENTIFICAZIO
NE";C$
40 IFLEN(C$)<>2THENPRINTCHR$(28)"DUE CAR
ATTERI PER FAVORE"CHR$(144):GOTO30
50 PRINT:PRINT"NOME",LEFT$(N$,16)
60 PRINT:PRINT"CODICE";C$
70 OPEN15,8,15
80 PRINT315,"I"
90 OPEN8,8,8,"$":GET#8,A$:CLOSE8
100 GOSUB500
110 IFW>20THEN200
120 PRINT"DISCO GIA' FORMATTATO."
130 PRINT"PREMI A QUANDO SEI PRONTO CO
N UN ALTRO DISCO."
140 PRINT"PREMI R PER RIFORMATTARE."
150 PRINT"PREMI X PER FINIRE."
160 GETI$:IFI$="A"THEN80
170 IFI$="R"THEN200
180 IFI$="X"THEN230
190 GOTO160
200 PRINT#15,"N0:"+N$","+C$
210 GOSUB500
220 PRINTW,X$,Y,Z
230 CLOSE15:END
500 INPUT315,W,X$,Y,Z
510 RETURN
```

............................Interprete PILOT

PILOT e' un linguaggio di programmazione molto piu' semplice del BASIC, che permettera' anche ai bambini molto piccoli di cominciare gia'a scrivere i loro primi programmi. E' particolarmente adatto ai quiz, ed anche per questo un programma in PILOT contenente dei quiz e' stato incluso in questo libro; ma l'immaginazione del bambino riuscira' senz'altro a trovare delle applicazioni piu' divertenti e interessanti. Ricordate: PILOT e' adatto, non per fare dei giochi per bambini, ma per insegnare ai bambini l'uso del computer (altrimenti, affermando cio', avrei aggiunto che programmare in PILOT e' piu' divertente che in BASIC, e voi avreste voluto creare dei programmi anche senza il mio aiuto).

............................Programmazione in PILOT

Ci sono due tipi di informazioni PILOT: le ISTRUZIONI e i COMANDI. Un'istruzione puo' essere usata come una parte del programma e non produce effetti se non dopo il RUN. Un comando no puo' essere usato come una parte di programma, ed e' sempre eseguito immediatamente (questo, come avrete gia' capito, in completo contrasto con il BASIC nel quale tutte le "parole" possono essere usate sia come istruzioni sia come comandi. Il BASIC, comunque, ha delle piccole differenze. INPUT e DATA non possono essere usati come

comandi, e LIST , sebbene possa essere usato come istruzione, provoca un brak del programma. Le istruzioni ed i comandi PILOT non possono essere confusi, poiche' un commando e' un'intera parola mentre un'istruzione si compone di una sola lettera (come nel BASIC, dove le parole di comando possono essere abbreviate premendo la lettera iniziale, seguita dal tasto SHIFT+ la seconda lettera. Non ci sono variazioni o eccezioni a questa regola. Le istruzioni, essendo di una lettera sola, non possono essere abbreviate).

C'e' una lista completa di tutti i comandi e le istruzioni PILOT nell'appendice in fondo al libro, ma prima di descrivere quelle piu' dettagliatamente, vorrei menzionare le altre caratteristiche del linguaggio che forse avreste preferito vedere incorporate al BASIC del 64:

1) La numerazione delle lettere e' automatica. I numeri di linea partono da 0 e si incrementano automaticamente. Quando un'istruzione viene immessa dopo il numero di linea, , essa viene inclusa nel programma e sullo schermo appare la linea successiva. Se un comando viene scritto dopo un numero di linea, esso viene eseguito, dopo che lo stesso numero di linea e' stato giudicato valevole.

2) Il controllo della sintassi e' immediato. Mentre nel BASIC la scritta "SINTAX ERROR" compare durante l'esecuzione di un programma, nel PILOT questo controllo avviene automaticamente e se la linea non e' sintatticamente esatta, non viene accettata nel programma.

...............................Comandi

Ci sono otto comandi nel PILOT, abbastanza facili da capire e per molti di essi praticamente identici ai comandi BASIC. Essi sono: NEW, RUN, LOAD, SAVE, LIST, LLIST, BASIC and EDIT.

I primi cinque non richiedono spiegazioni. LLIST e' un comando trovabile in molte versioni BASIC (ma non sul Commodore 64) che lista il programma in memoria tramite

una stampante. BASIC vi fa uscire dall'interprete PILOT per tornare al linguaggio tradizionale. EDIT, che deve essere seguito da un numero di linea, causa l'immediata interruzione dell'auto-numerazione, e visualizza la linea richiesta. Non e' possibile scrivere sopra la linea sbagliata ma occorre riscriverla totalmente. E' permesso solo aggiungere linee di programma, non cancellarle o inserirle, sebbene le linee non volute possano essere cancellate grazie all'uso dell'istruzione "*" (vedere piu' avanti).

................................Istruzioni

Ci sono sei istruzioni valide per un programma in PILOT. Esse sono:

T J C A M e *

T (come Type) equivale al comando PRINT in BASIC, eccetto che nel testo non c'e' bisogno di includere gli apici.

J (come Jump) equivale al GOTO del BASIC.

C (come Clear Screen) pulisce lo schermo e assegna i colori dello sfondo; equivale a PRINTCHR$(17) e POKE

53281,n.

A (come Answer) equivale ad INPUT.

* equivale a REM.

M (come Match) non ha un esatto equivalente in BASIC. Quello che fa e' definire una risposta "preferita". Quindi, ogni volta che un'istruzione A viene incontrata, e l'operatore immette un valore in memoria (presumibilmente una risposta ad una domanda appena fatta tramite T),

questo valore viene confrontato con la risposta preferita. Se sono identiche compare un "flag", altrimenti un "flag" viene cancellato. PILOT cerca la traccia del "flag status" e ne fa uso quando interpreta istruzioni condizionate.

..............................Condizioni

Il PILOT possiede due condizioni, Y e N (per "Yes" e "No", ma se voi preferite, Vero e Falso). Ciascuna di queste due istruzioni puo' essere "appesa" (vedere pu' avanti) in una istruzione T o J, nel qual caso l'istruzione verra' esguita solo in condizioni ben determinate. Ad esempio, l'istruzione:

TY LA VOSTRA RISPOSTA E' ESATTA

visualizzera' la frase SOLO quando il "flag" e' inserito, e questo naturalmente avverra' solo dopo che la risposta dell'operatore ad un'istruzione A e' stata confrontata e trovata identica alla risposta preferita dall'istruzione M.

......................Il Punto e virgola

Ogni istruzione-lettera DEVE essere seguita immediatamente da un "appendice". Ci sono solo tre possibili "appendici": le due condizioni (Y e N) menzionate prima e ";" (il punto e virgola)

Le istruzioni C, A e M devono sempre essere seguite da un punto e virgola. Le istruzioni T e J possono essere seguite da un punto e virgola (se non sono condizionate) o da una Y o una N. L'istruzione *, strettamente parlando, dovrebbe essere seguita da un punto e virgola, ma poiche' rappresenta la medesima funzione che svolge la parola REM nel BASIC, facendo ignorare al programma cio' che segue, e' sintatticamente indifferente scrivere dopo di esso.

Il punto e virgola non ha una funzione specifica se non quella di rendere tutte le istruzioni PILOT, di controllo

o altro, lunghe due caratteri. Sebbene le istruzioni PILOT debbano considerarsi corrette solo quando si riferiscono ad una singola lettera, puo' essere utile per la vostra memoria avere sottomano l'intera serie formata dai comandi in due caratteri:
T; TY TN J; JY JN C; A; M; *;

.......................Messaggi d'errore

Ci sono solo cinque possibili tipi di errore, che si identificano con i seguenti messaggi:

SINTAX ERROR: avete dimenticato un comando o sbagliato a scrivere un'istruzione. In entrambi i casi cio' che avete digitato viene ignorato e siete messi in grado di riscrivere correttamente il vostro messaggio. Un errore di sintassi non puo' essere visualizzato mentre un programma in PILOT sta girando.

A WITHOUT M ERROR: Cosi' come nel BASIC siete avvertiti quando il programma incontra un'istruzione NEXT senza un precedente FOR, nel PILOT il computer vi fa notare che avete usato una A senza una precedente M. Comunque voi potete avere diverse A seguenti una singola M. Una volta che una risposta pre-determinata e' stata creata tramite un'istruzione M, essa rimane valida senza successive istruzioni M.

UNDEFINED LINE ERROR: Viene provocato dalla poco piacevole auto-numerazione del PILOT. Comunque, vedrete quest'errore durante la programmazione se volete visualizzare una linea inesistente, o durante il corso di un programma, se un'istruzione J e' seguita da un numero di linea superiore alla fine del programma.

OUT OF MEMORY ERROR: Il programma ha raggiunto il massimo delle linee digitabili (100 linee).

INVALID COLOUR ERROR: Avete digitato una linea nella quale un'istruzione C e' seguita da un numero al di fuori

del raggio 0-15, o uguale a 13 (che e' riservato al colore di stampa). In verita' questo non e' altro che un particolare tipo di SINTAX ERROR, e per questo non puo' essere visualizzato mentre un programma gira.

Nota importante: sebbene questi siano solo errori ricorrenti tramite il PILOT, e' possibile incorrere nei consueti errori BASIC. Un esempio degli errori ricorrenti:

Sintax error: avete commesso un errore nel digitare l'interprete PILOT.

Break error: avete premuto inavvertitamente il tasto RUN/STOP (questo tasto puo' essere disabilitato - vedete le note del programma Auto-line).

Device not present: avete provato ad usare il comando LLIST senza avere la stampante accesa (o LOAD o SAVE senza avere la periferica corrispondente, a nastro o disco, inserita - guardate il paragrafo "Note del programma" piu' in basso).

Se qualcuno di questi errori dovesse ricorrere (e riconoscerete se errori BASIC o PILOT dalla presenza del familiare messaggio READY), vi troverete fuori dall'interprete PILOT, nel Basic di tutti i giorni. Prevedendo questa situazione, per rientrare nell'interprete stesso senza cancellare il programma in PILOT ancora in memoria, bastera' digitare "GOTO 100".

............................E per cominciare...

Voi ora sapete ogni cosa che vi puo' servire nella programmazione PILOT. Se volete saperne di piu', digitate e fate girare il programma "Quiz di geografia", e tutto vi sara' piu' chiaro. Quindi cominciate subito a scrivere i vostri programmi in PILOT e rendetevi conto di quanto sia facile farlo.

.........................Note del programma.

Tutte le parti importanti dell'"Interprete Pilot" sono corredate da istruzioni REM, cosi' da non avere difficolta' a capire come lavora il programma. Le opzioni LOAD e SAVE (linee 6000-6080 e 7000-7070) nel listato sono per i possessori di drive. Per chi possiede solo il registratore, e' necessario cambiare le seguenti due linee:

6020 OPEN 8,1,1,"PILOTPROG"

7010 OPEN 8,1,0,"PILOTPROG"

(entrambi i possessori possono elaborare queste semplici istruzioni in modo tale da permettere di chiamare con nomi individuali i programmi in Pilot)

Solo altri due appunti per aiutarvi nella digitazione del programma:

I messaggi di introduzione fasulli (linee 20 e 30) possono essere presi direttamente dai reali messaggi introduttori che appaiono all'accensione del Commodore 64. Posizionate il cursore prima di ogni messaggio, inserite un numero di linea e il messaggio entrera' a far parte del vostro programma. Non e' necessario, naturalmente, usare lo stesso stile della Commodore; se state scrivendo un programma per bambini, perche' non inserire il loro nome in un messaggio di introduzione?

I caratteri grafici nelle linee 120-190 sono , in ogni caso, il carattere shiftato sul tasto rappresentante la seconda lettera del particolare comando PILOT (questo e' particolarmente importante alla linea 180, dove SHIFT+L e' quasi identico alla normale lettera "L").

```
5 REM *** PRG.#23 -PILOT INTERPRETE ***
10 POKE53280,13:POKE53281,5:PRINTCHR$(14
7);CHR$(153)
20 PRINT"     **** COMMODORE 64 PILOT V0
****"
```

```
30 PRINT:PRINT" 64K RAM SYSTEM    100 PI
LOT LINES FREE"
40 DIMA$(100)
100 N=L
110 PRINTN;TAB(6);:INPUTI$
120 IFI$="NEW"ORI$="N¯"THENRUN
130 IFI$="RUN"ORI$="R╱"THEN1000
140 IFI$="LIST"ORI$="L¬"THEN2000
150 IFI$="LLIST"ORI$="LL"THEN3000
160 IFI$="BASIC"ORI$="B♠"THENEND
170 IFI$="SAVE"ORI$="S♠"THEN6000
180 IFI$="LOAD"ORI$="L┌"THEN7000
190 IFLEFT$(I$,2)="E¯"THENI$="EDIT"+MID$
(I$,3)
200 IFLEFT$(I$,4)="EDIT"THEN4000
210 J$=LEFT$(I$,1)
220 IFJ$="T"ORJ$="J"ORJ$="A"ORJ$="M"ORJ$
="*"ORJ$="C"THEN5000
230 GOTO9010
1000 REM RUN
1010 MA$=""
1020 FORJ=0TOL-1
1030 J$=LEFT$(A$(J),1):IFJ$=""THEN1900
1040 IFJ$="*"THEN1900
1050 K$=MID$(A$(J),2,1)
1060 P$=MID$(A$(J),3)
1070 IFJ$="T"THEN1200
1080 IFJ$="A"THEN1300
1090 IFJ$="M"THEN1400
1100 IFJ$="J"THEN1500
1110 IFJ$="C"THEN1600
1200 REM T
1210 IFK$=";"THEN1700
1220 IF(K$="Y"ANDF=-1)OR(K$="N"ANDF=0)TH
EN1700
1230 GOTO1900
1300 REM A
1310 INPUTAN$
1320 IFMA$=""THEN9020
1330 F=AN$=MA$
1340 GOTO1900
1400 REM M
1410 MA$=P$
1420 GOTO1900
1500 REM J
1510 IF(K$=";")OR(K$="Y"ANDF=-1)OR(K$="N
"ANDF=0)THEN1550
1520 GOTO1900
1550 J=VAL(P$)
1560 IFJ<0ORJ>L-1ORJ<>INT(J)THEN9030
1570 J=J-1:GOTO1900
1600 REM C
1610 C=VAL(P$):POKE53281,C
1620 P$=CHR$(147)
1700 PRINTP$
1900 NEXT
1910 GOTO100
2000 REM LIST
2010 FORJ=0TOL-1
2020 PRINTJ;TAB(8)A$(J)
2030 NEXT
2040 GOTO100
3000 REM LLIST
3010 OPEN4,4
3020 FORJ=0TOL-1
3030 PRINT#4,J;TAB(8)A$(J)
3040 NEXT
```

```
3050 CLOSE4
3060 GOTO100
4000 REM EDIT
4010 N=VAL(MID$(I$,5))
4020 IFN>L-1ORN<0ORN<>INT(N)THEN9030
4030 PRINTN;TAB(8);A$(N)
4040 GOTO110
5000 REM SYNTAX
5010 IFJ$="*"THEN5100
5020 K$=MID$(I$,2,1)
5030 IFK$=":"THEN5050
5040 IF(J$<>"T"ANDJ$<>"J")OR(K$<>"Y"ANDK
$<>"N")THEN9010
5050 IFJ$<>"C"THEN5100
5060 C=VAL(MID$(I$,3))
5070 IFC<0ORC>15ORC<>INT(C)ORC=13THEN905
0
5100 IFL>100THEN9040
5110 A$(N)=I$
5120 IFL=NTHENL=L+1
5130 GOTO100
6000 REM SAVE (DISC VERSION)
6010 IFL=0THEN100:REM NO PROG TO SAVE
6020 OPEN8,8,8,"@0:PILOTPROG,S,W"
6030 PRINT#8,L
6040 FORJ=0TOL-1
6050 PRINT#8,A$(J)
6060 NEXT
6070 CLOSE8
6080 GOTO100
7000 REM LOAD (DISC VERSION)
7010 OPEN8,8,8,"0:PILOTPROG,S,R"
7020 INPUT#8,L
7030 FORJ=0TOL-1
7040 INPUT#8,A$(J)
7050 NEXT:CLOSE8
7060 CLOSE8
7070 GOTO100
9000 REM ERRORS
9010 PRINT"SYNTAX ERROR":GOTO100
9020 PRINT"A WITHOUT M ERROR":GOTO100
9030 PRINT"UNDEFINED LINE ERROR":GOTO100
9040 PRINT"OUT OF MEMORY ERROR":GOTO100
9050 PRINT"INVALID COLOUR ERROR":GOTO100
```

................Quiz di Geografia (PILOT)

La persona che ha scritto questo quiz conosce la geografia molto meglio di me.

Vi ricordo che lo scopo del programma non e' tanto quello di saggiare la vostra cultura, quanto quello di mostrarvi il piu' semplicemente possibile le caratteristiche dell'interprete PILOT.

```
0   C;0
1   T;QUIZ DI GEOGRAFIA
2   T;
3   T;1 - INGHILTERRA
4   T;2 - U.S.A.
5   T;
6   T;PREGO, SCEGLI
7   M;1
8   A;
9   JN36
10  M;NO
11  T;MANCHESTER E' NEL LANCASHIRE
12  A;
13  TNNO - PIU' GRANDE MANCHESTER
14  T;E' LIVERPOOL NEL LANCASHIRE
15  A;
16  TNNO - MERSESYDE
17  T;E' LANCASTER NEL LANCASHIRE
18  A;
19  TYSI' LO E'
20  T;QUAL E' LA CAPITALE DEL NORFOLK
21  M;NORWICH
22  A;
23  TNE' NORWICH
24  T;QUAL E' LA CAPITALE DEL SUSSEX
25  A;
26  T;NON C'E' ALCUN PAESE
27  T;IN QUALE PAESE E' CANTERBURY
28  M;KENT
29  A;
30  TNNO - KENT
31  T;
32  T;VUOI PROVARE CON GLI STATI UNITI
33  M;SI'
34  A;
35  JN70
36  T;QUAL E' LA CAPITALE DELLA FLORIDA
37  M;TALLAHASSEE
38  A;
39  JY43
40  T;INDIZIO - 2 'L'S - 2 'S'S - 2 'E'S
```

```
41  A;
42  TNTALLAHASSEE
43  T;NEW YORK E' LA CAPITALE DELLO STATO DI NEW YORK
44  M;NO
45  A;
46  TYCORRETTO MA
47  TNNO LO E'
48  T;QUAL E'
49  M;ALBANY
50  A;
51  TNE' ALBANY
52  M;SI'
53  T;CARSON CITY E' LA CAPITALE DEL NEVADA
54  A;
55  TNLO E'
56  T;E' ST.JOHN LA CAPITALE DEL MINNESOTA
57  A;
58  TYNO NON LO E'
59  TNGIUSTO
60  T;LA CAPITALE E' ST. PAUL
61  T;QUAL E' LA CAPITALE DEL TEXAS
62  M;AUSTIN
63  A;
64  TNDOVETE CORREGGERE PRIMA DI CONTINUARE
65  JN61
66  T;
67  T;FINE DEL QUIZ
68  T;
69  M;SI'
70  T;DAI ANCORA RUN
71  A;
72  JYO
73  C;5
```

................................Contando

Quanto fa

```
 13 +
 15
-----
  ?
```

(Questo e' un programma per bambini molto piccoli)

Molti bambini troverebbero difficile rispondere alla domanda quando scritta in quella forma, ma non avrebbero problemi se trovassero 13 gettoni e 15 gettoni con la richiesta di addizionarli. Questo programma permette a chi lo usa di rispondere correttamente in due modi; dapprima la somma viene visualizzata come sopra, e quindi, se viene inserita una risposta sbagliata, i numeri vengono mostrati come due file di quadrati colorati. Si sente inoltre un suono ad ogni quadrato che appare, cosi' la risposta puo' essere data o contando i quadrati o i suoni emessi.

.......................Note del programma

La difficolta' della somma puo' essere variata alterando la linea 30, sebbene raggiungendo una somma superiore a venti sarete costretti a modificare l'uscita sul video. Come la linea stabilisce, il primo numero apparira' sempre nel raggio 11-20, mentre il secondo nel raggio 1-20; la risposta comportera' sempre la pressione di due tasti.

Il ciclo di ritardo alla linea 690 e 730 determina la velocita' alla quale i quadrati colorati vengono POKEati sullo schermo, velocita' che puo' essere variata a seconda dell'abilita' del bambino.

Non ho incluso alcun tipo di algoritmo per il calcolo del punteggio. Potreste aggiungere una routine che assegni un punteggio per le risposte, un punto per ogni somma esatta ed un "bonus" a chi non ricorre mai all'ausilio dei quadrati colorati (non dimenticate di modificare l'istruzione RUN alla linea 430 con un GOTO, per evitare che ad ogni nuova domanda venga cancellato il punteggio).

```
5 REM*** PRG.#25- CONTANDO ***
10 POKE53280,15:POKE53281,1:PRINTCHR$(14
7);CHR$(31)
20 X=RND(-TI):D=54272:FORJ=1TO39:B$=B$+C
HR$(32):NEXT
30 X=INT(RND(1)*11)+10:Y=INT(RND(1)*20)+
1:Z=X+Y
40 PRINTTAB(17)"QUANTO FA":PRINT
50 N=X:GOSUB500
60 PRINT"+"
70 N=Y:GOSUB500
80 PRINT:PRINTTAB(19);CHR$(101);CHR$(101
);CHR$(30)
90 PRINTTAB(17):INPUTA$
100 A=VAL(A$):IFA=ZTHENPRINTCHR$(156);CH
R$(17);TAB(16);"ESATTO":GOTO400
110 IFFTHEN300
120 PRINTCHR$(17);CHR$(28):PRINT"PROVA I
N QUESTO MODO":PRINT
130 PRINT:PRINTTAB(18)X
140 N=X:P=1542:GOSUB600
150 PRINT:PRINTTAB(18)Y
160 N=Y:P=1622:GOSUB600
170 PRINTCHR$(30):FORJ=1TO10:PRINTCHR$(1
45);:NEXT:
180 F=-1:GOTO90
300 PRINTCHR$(129):PRINTB$:PRINT"NO. LA
RISPOSTA ERA"Z
400 FORJ=1TO8:PRINTB$:NEXT
410 PRINTCHR$(158)TAB(2)"██PREMI LO SPAZ
IO PER UN'ALTRA DOMANDA"
420 GETA$:IFA$<>CHR$(32)THEN420
430 RUN
500 P$=STR$(N):IFLEN(P$)=2THENP$=CHR$(32
)+P$
510 PRINTTAB(18)CHR$(144);P$;
520 RETURN
600 FORJ=1TON
610 POKE54296,15
620 POKE54277,15
630 POKE54278,15
640 POKE54273,20+J
650 POKE54272,20+J
660 POKE54276,17
670 POKEP+J*2,160
680 POKEP+J*2+D,INT(RND(1)*12)+2
690 FORK=1TO100:NEXT
700 POKE54276,0
710 POKE53277,0
720 POKE54278,0
730 FORK=1TO100:NEXT
740 NEXT
750 RETURN
```

Non c'e' niente di piu' noioso di un manuale che parli di un computer. Volete veramente che vi spieghi cosa sia un byte? Sono sicuro di no. Comunque, perche' spiegarvi cos'e' quando avete un computer che ve lo puo' mostrare?

Date RUN al programma e cosi' vedrete... UN BYTE.

Il byte assumera' qualunque valore che avrete selezionato, ad eccezione dei valori esterni al raggio 0-255, nel qual caso si avra' un valore casuale. E sotto il byte comparira' l'esatta spiegazione di come il byte rappresenti quel valore.

Semplice, non e' vero? Voi ora sapete tutto cio' che c'e' da sapere sui bytes. Chi ha piu' bisogno di manuali?

```
5 REM*** PRG.#26- BYTE ***
10 FORJ=1TO3:B$(0)=B$(0)+CHR$(166):NEXT
20 B$(1)=CHR$(28)+CHR$(18):FORJ=1TO3:B$(
1)=B$(1)+CHR$(160):NEXT
30 B$=B$+CHR$(146)+CHR$(144)
40 PRINTCHR$(147);CHR$(144)
```

```
50 POKE53280,7:POKE53281,1
60 INPUT"INSERISCI UN NUMERO <0-255>";N
70 N=INT(N):IFN<0ORN>255THENN=INT(RND(1)
*256):
80 Z=0:FORJ=7TO0STEP-1
90 PRINTCHR$(19);CHR$(17);CHR$(17)
100 T=(9-J)*4-4
110 PRINTTAB(T)"BIT":PRINTTAB(T)J:PRINT
120 PRINTTAB(T-1);2↑J
130 B(J)=INT(N/2↑J):N=N-B(J)*2↑J
140 PRINTTAB(T);B$(B(J))
150 PRINTCHR$(19):FORK=15TOJSTEP-1:PRINT
CHR$(17);:NEXT
160 Z=Z+B(J)*2↑J:Z$=STR$(B(J)*2↑J)
170 PRINTB(J)"* 2 ↑ J";TAB(10)"="TAB(16-
LEN(Z$));Z$
180 NEXT
190 PRINTCHR$(17)"TOTALE"TAB(16-LEN(STR$
(Z)));Z
200 PRINTCHR$(17)"PREMI UN TASTO PER CON
TINUARE"
210 GETZ$:IFZ$=""THEN210
220 GOTO40
```

......................Bits

Nel precedente programma avete visto come un byte rappresenti un numero intero da 0 a 255. Questo programma vi permette di cambiare il valore di un byte, accendendo e spegnendo i suoi bits individuali.

I bits sono numerati convenzionalmente da destra a sinistra, ed ogni bit ha il seguente valore:

Numero del Bit	Valore (acc.)	Valore (spento)
0 (estrema des.)	1	0
1	2	0
2	4	0
3	8	0
4	16	0
5	32	0
6	64	0
7 (estrema sin.)	128	0

Il byte viene primo visualizzato con il valore totale a 0 (tutti i bits spenti). Usa i tasti da 0 a 7 per accendere ogni bit, e CTRL+ i tasti da 0 a 7 per spegnere i bits.

Vedete se riuscite ad ottenere ogni possibile valore del byte (0-255) accendendo e spegnendo i bits corretti.

```
5 REM *** PRG.# 27-BITS ***
10 FORJ=0TO7:READA(J):NEXT
20 DATA146,144,5,28,159,156,30,31
30 PRINTCHR$(147);CHR$(144)
40 POKE53280,6:POKE53281,1
50 P=1516:D=54272
60 FORJ=PTOP+14STEP2
70 POKEJ,160:POKEJ+D,13
80 NEXT
90 PRINTCHR$(19)TAB(58)"     "CHR$(19)TA
B(58)B:REM 5 SPACES TO COVER NUMBER.
100 GETI$:IFI$=""THEN100
110 A=ASC(I$)
120 IFA>47ANDA<56THENC=2:A=A-48:GOTO160
130 FORJ=0TO7
140 IFA(J)=ATHENC=13:A=J:J=10
150 NEXT:IFJ=8THEN100
160 POKEP+D+2*(7-A),C
170 B=0:FORJ=7TO0STEP-1
180 B=B+(2↑JAND(PEEK(P+D+2*(7-J))AND15)=
2)
190 NEXT
200 GOTO90
```

..........................And/Or

Se pensassi che voi non siate realmente interessati in questo studio dei bits e dei bytes, non includerei questo programma. Comunque voi avete acquistato u Commodore 64, il solo computer al mondo che ritenga che il linguaggio BASIC si risolva interamente nella parola POKE, e cosi' il concetto di "bit-level" del computing e' probabilmente piu' importante per voi che per altri utenti di computer.

Molti manuali di computer non fanno nemmeno cenno alle condizioni logiche di AND e OR, mentre il manuale d'uso del Commodore 64 ne introduce l'uso all'inizio del capitolo 5. Il manuale tratta l'uso della POKE 53280,C per cambiare il colore al bordo. Come sapete ogni valore da 0

a 15 puo' essere POKEato nell'indirizzo 53280, ma questo indirizzo, come ogni altro singolo byte, puo' contenere un valore compreso tra 0 e 255. Se voi chiedete PEEK(53280), il valore di ritorno non sara' nel raggio da 0 a 15, sebbene voi abbiate appena POKEato un valore di quel tipo. L'espressione esatta per ottenere i valori nel corrente codice di colori non e' PEEK(53280) ma PEEK(53280)AND15.

La ragione di cio' e' spiegata dal fatto che solo i "valori bassi" (i bits da 0 a 3) dell'indirizzo 53280 sono usati per memorizzare il codice del colore-bordo; i "valori alti" (da 4 a 7) hanno altri scopi. Per questo, per ottenere un valore dell'indirizzo 53280 che vari da 0 a 15 occorre disabilitare i valori alti, e questo viene fatto dall'istruzione logica AND.

Usando AND con due numeri si ottiene un valore che, quando espresso in bits, possiede solo i bits accesi che risiedono in un numero E nell'altro.

Usando OR con due numeri si ottiene un valore che, quando espresso in bits, mostra i bits accesi in un numero O nell'altro.

Per esempio, immaginiamo che PEEK(53280) ci dia un valore di 250. Espresso in bits (in numeo binario), diventa 11111010. Se questo ha un AND con 15 (binario = 00001111), il risultato sara' il numero (sempre binario) 00001010, che non e' altro che il numero decimale 10.

250 AND 15 = 10

Il byte 000111 e' stato usato come una maschera per permettere di ignorare, qualunque valore fosse ritornato dalla POKE(53280), il valore alto.

Come facciamo a dare un senso a tutto cio'? Fate girare il programma e tutto vi diventera' chiaro. La parte sinistra dello schermo vi mostrera' il risultato di un AND tra due bytes, e la parte destra il risultato di un OR tra gli stessi due bytes. Voi potete accendere e spegnere i

bits individuali del byte principale allo stesso modo dei precedenti programmi (con l'uso dei tasti da 0 a 7, col tasto CTRL premuto per resettarli). Per accendere e spegnere i bits del byte mascherato, usate il tasto-funzione F3, quindi i tasti 0-7 con CTRL come prima. Per ritornare all'originale byte, usate invece il tasto-funzione F1.

```
5 REM ***PRG.# 28- AND/OR ***
10 FORJ=0TO7:READA(J):NEXT
20 DATA146,144,5,28,159,156,30,31
30 FORJ=1TO39:B$=B$+CHR$(32):NEXT
40 PRINTCHR$(147);CHR$(144)
50 POKE53280,6:POKE53281,1
60 P(0)=1146:P(1)=1167:P(2)=1386:P(3)=1407:P(4)=1626:P(5)=1647:D=54272
70 FORK=0TO5:FORJ=P(K)TOP(K)+14STEP2
80 POKEJ,160:POKEJ+D,13
90 NEXTJ,K
100 PRINTCHR$(19):FORJ=1TO5:PRINTCHR$(17);:NEXT:PRINTB$
110 PRINTCHR$(145);TAB(4);B(0)"AND"B(2);TAB(24);B(0)"OR"B(2)
120 FORJ=1TO5:PRINTCHR$(17);:NEXT:PRINTB$
130 PRINTCHR$(145);TAB(5);"="B(0)ANDB(2);TAB(25);"="B(0)ORB(2)
140 FORJ=1TO5:PRINTCHR$(17);:NEXT:PRINT"PREMI F1 PER MODIFICARE IL MAIN BYTE"
150 PRINT"PREMI F3 PER MODIFICARE IL MASKING BYTE"
160 PRINT"PREMI IL NUM. PER ATTIVARE IL BIT"
170 PRINT"PREMI CTRL E IL NUM. PER RESETT. IL BIT"
180 GETI$:IFI$=""THEN180
190 A=ASC(I$)
200 IFA=133THENP=0:GOTO180
210 IFA=134THENP=2:GOTO180
220 IFA>47ANDA<56THENC=2:A=A-48:B(P)=B(P)OR(2↑A):GOTO260
230 FORJ=0TO7
240 IFA(J)=ATHENC=13:A=J:B(P)=B(P)ANDNOT(2↑A):J=10
250 NEXT:IFJ=8THEN180
260 A=D+2*(7-A)
270 POKEP(P)+A,C
280 POKEP(P+1)+A,C
290 C=13:IF((15ANDPEEK(P(0)+A))=2)AND((15ANDPEEK(P(2)+A))=2)THENC=2
300 POKEP(4)+A,C
310 C=13:IF((15ANDPEEK(P(1)+A))=2)OR((15ANDPEEK(P(3)+A))=2)THENC=2
320 POKEP(5)+A,C
330 GOTO100
```

.......................CHR Set

Questo programma ci mostra il modo nel quale il set dei caratteri del Commodore 64 viene memorizzato nella memoria del computer, producendo dele caselle 8x8 per ogni carattere (uno studio di come cio' avvenga e' indispensabile per quando vorrete creare i vostri propri caratteri grafici).

Ci sono in totale 512 caratteri, 256 nel set grafico e 256 nel set testo (alcuni caratteri si ripetono in entrambi i set), e questi sono memorizzati in un ordine determinato dal loro valore POKE, con i caratteri del set testo immagazzinati dopo i caratteri grafici.

Ogni carattere e composto da 4 bits (cioe' 8 bytes), e ogni bit e' rappresentato da un punto sullo schermo. Se un bit e' acceso (=1) il punto compare nel colore di stampa, se spento (=0) nel colore dello sfondo. E' possibile visualizzare un carattere perche' esso non e' rappresentato da un bit ma da un'intera griglia di bits; spegnendo il bit esso diventera' bianco, accendendolo diverra' nero, formando la figura del carattere.

......................Note del programma

I set dei caratteri sono locati nella CHRGEN ROM, che occupa 4096 indirizzi consecutivi (512 caratteri, ognuno richiedente 8 bytes) a partire dall'indirizzo 53248. La gestione video e suono del Commodore 64 (VIC e SID) usa gli stessi indirizzi di memoria, e il computer riconosce la CHRGEN ROM o il VIC e SID tramite l'accensione di un singolo bit del byte numero 1. Se il bit numero 2 di questo byte e' acceso, allora sono i chips VIC e SID che occupano queste locazioni, mentre se e' spento e la CHRGEN ROM. Come avrete scoperto dal programma precedente, i bits individuali possono essere accesi o spenti dalla maschera,

usando AND e OR. La linea 30 con l'uso di AND e PEEK(1) con 251 spegne il bit numero due lasciando inalterati gli altri bits. La linea 50 fa esattamente il contrario, grazie a OR e PEEK(1) con 4.

Tutto cio' ci rende molto difficile la lettura della CHRGEN ROM, ma questa difficolta' puo' essere superata facendo una copia esatta della CHRGEN ROM in un'altra zona di memoria del computer. La linea 40 copia l'intero set di caratteri dalla ROM all'indirizzo 12288-16383 (la parte alta dei primi 16k di memoria RAM). Una volta fatto cio', e' facile tramite delle PEEK visionare ogni carattere (linea 160).

Notate che, sebbene il computer impieghi un certo tempo per effettuare questa operazione, e' necessario eseguirla una volta soltanto. In seguito bastera' digitare RUN 100 per far ripartire il programma.
(Altre caratteristiche di questo programma vi saranno spiegate piu' avanti nel programma "Generatore di caratteri").

```
3 REM ***PRG.#29 - CHR SET ***
5 POKE53281,6:PRINTCHR$(147);CHR$(5);"PE
R FAVORE ATTENDERE UN PO' DI TEMPO"
10 POKE52,48:POKE56,48
20 POKE56334,PEEK(56334)AND254
30 POKE1,PEEK(1)AND251
40 FORJ=0TO4095:POKE12288+J,PEEK(53248+J
):NEXT
50 POKE1,PEEK(1)OR4
60 POKE56334,PEEK(56334)OR1
70 POKE53272,(PEEK(53272)AND240)+12
100 POKE53280,3:POKE53281,3:PRINTCHR$(14
7);CHR$(144)
110 S=1240:D=54272
120 C=12288
130 FORJ=0TO511:PRINTCHR$(147);:G=J:IFJ>
255THENG=J-256:PRINTCHR$(14);
140 PRINTG:POKE1034,G:POKE1034+D,0
150 FORK=0TO7
160 P=PEEK(C+8*J+K):PRINTP;
170 FORL=0TO7
180 Q=INT(P/2↑(7-L)):P=P-Q*2↑(7-L)
190 POKES+L+40*K,160
200 POKES+D+L+40*K,ABS(Q=0)
210 NEXTL,K
220 PRINT:PRINT"PREMI ▪SPAZIO▪ PER CONTI
NUARE"
230 GETI$:IFI$<>CHR$(32)THEN230
240 NEXTJ
300 PRINTCHR$(142)
310 REM USARE 'RUN 100' PER RIPART. UNA
VOLTA CHE IL SET-CARATTERI E' DEFINITO
```

......................Hex tables

Nel tentativo di aiutarvi, come farvi imparare alcune tavole di moltiplicazione? Non le normali tavole decimali, naturalmente, ma quelle esadecimali.

Selezionate quale tavola desiderate vedere (0-F), e quindi usate "piu' grande di" e "piu' piccolo di" (i tasti < e > SHIFTati e non) per far comparire le tavole adiacenti.

```
5 REM ***PRG.#30-HEX TABLES ***
10 POKE53280,0:POKE53281,0:PRINTCHR$(147
);CHR$(5)
20 H$="0123456789ABCDEF"
30 V$="0123456789:;<=>?"
40 PRINT"QUALE TAVOLA HEX (0-F)"
50 GETI$
60 IF(I$<"0"ORI$>"9")AND(I$<"A"ORI$>"F")
THEN50
70 FORJ=0TO15:IFI$=MID$(H$,J+1,1)THENA=J
80 NEXT
90 PRINTCHR$(147):POKE532812
100 POKE53280,3:POKE53281,3:PRINTCHR$(14
7);CHR$(144)
110 S=1240:D=54272
120 C=12288
130 FORJ=0TO511:PRINTCHR$(147);:G=J:IFJ>
255THENG=J-256:PRINTCHR$(14);
140 PRINTG:POKE1034,G:POKE1034+D,0
150 FORK=0TO7
160 P=PEEK(C+8*J+K):PRINTP;
170 FORL=0TO7
180 Q=INT(P/2↑(7-L)):P=P-Q*2↑(7-L)
190 POKES+L+40*K,160
200 POKES+D+L+40*K,ABS(Q=0)
210 NEXTL,K
220 PRINT:PRINT"PREMI SPAZIO PER CONTI
NUARE"
230 GETI$:IFI$<>CHR$(32)THEN230
240 NEXTJ
300 PRINTCHR$(142)
310 REM USARE 'RUN 100' PER RIPART. UNA
VOLTA CHE IL SET-CARATTERI E' DEFINITO
```

......................Mempeek

Non c'e' miglior modo di conoscere il vostro Commodore 64 se non quello di usare il comando PEEK per perlustrare la sua memoria. In molte circostanza, comunque, quando si

agisce all'interno dell'area di programmazione Basic o nella memoria di schermo i numeri visualizzati sarebbero maggiormente utili se mostranti il carattere ASCII che rappresentano.

Questo programma vi permette di selezionare ogni indirizzo di partenza, per usare PEEK fino a 23 bytes consecutivi alla volta, mostrando sia il valore PEEK normale che il corrispondente carattere ASCII.

Non appena un blocco di 23 indirizzi viene visualizzato, avete la possibilita' di usare PEEK per gli stessi 23 indirizzi (premendo R), continuare per il successivo blocco di 23 (premendo C), selezionare un nuovo indirizzo di partenza per la lettura (premendo N) o terminare il programma (premendo X).

......................Note del programma

La maggior parte del programma lavora per tradurre i valori PEEK nei caratteri ASCII. Questo costituisce un problema, perche' non tutti i caratteri ASCII possono essere rappresentati sullo schermo. Alcuni di essi rappresentano caratteri di controllo, che cancellerebbero lo schermo, cambierebbero il colore di stampa o commutando il computer nel modo-testo.

Le istruzioni DATA alle linee 1000-1200 contengono tutti i codici ASCII che intralcerebbero la visione dello schermo, e alle linee 2000-2220 ci sono le spiegazioni testuali di questi codici, che vengono visualizzate al loro posto (linea 810).

```
5 REM*** PRG.# 31- MEMPEEK ***
10 GOSUB900
20 POKE53280,11:POKE53281,12:PRINTCHR$(1
47);CHR$(144)
100 INPUT"INDIRIZZO DI PARTENZA";S
110 PRINTCHR$(147);CHR$(28);"INDIRIZZO",
"PEEK","STAMPA COME"CHR$(144)
120 FORJ=STOS+22
130 P=PEEK(J)
140 C$=CHR$(P):GOSUB800
```

```
150 PRINTJ,P,C$
160 NEXT
170 PRINTCHR$(5)"C=CONT: R=RIPETE: N=NUO
VO IND.: X=ESCI"CHR$(144);
180 GETI$
190 IFI$="C"THENS=S+23:GOTO110
200 IFI$="R"THEN110
210 IFI$="N"THEN20
220 IFI$="X"THENEND
230 GOTO180
800 IFP$(P)=""THENRETURN
810 C$=P$(P):IFP=14THENC$=C$+CHR$(142):R
EM RESTA IN MODO GRAFICO
820 FORK=1TO16
830 IFT(K)=PTHENK=20
840 NEXT:IFK=17THENRETURN
850 CC$=CHR$(P):IFP=152THENCC$=CHR$(151)
:REM MED.GRAY TO DARK GRAY
860 C$=CC$+C$+CHR$(144)
870 RETURN
900 DIMT(30),P$(255)
910 RESTORE:FORJ=1TO30:READT(J):NEXT
920 FORJ=1TO30:READP$(T(J)):NEXT
930 RETURN
1000 DATA5,28,30,31,129,144,149,150,151,
152,153,154,155,156,158,159
1100 DATA17,29,145,157
1200 DATA14,142,8,9,18,146,19,147,13,141
2000 DATABIANCO,ROSSO,VERDE,BLU,ARANC.,N
ERO,MARR.,ROSSO C,GRIGIO S,GRIGIO M
2010 DATAVERDE C,BLU C,GRIGIO C,VIOLA,GI
ALLO,AZZURRO
2100 DATACURSORE GIU',CURSORE DESTRA,CUR
SORE SU',CURSORE SINISTRA
2200 DATAMODO TESTO,MODO GRAFICO,DISABIL
ITA MODO,RIABILITA MODO
2210 DATAREVERSE ON,REVERSE OFF,HOME,PUL
ISCE SCHERMO
2220 DATARETURN,SHIFT+RETURN
```

...........................Demo Sort

Questa e' un'affascinante ed ipnotica dimostrazione della routine "Sort" in azione.

Sarei capace di ammirare questo programma per ore intere (e lo faccio!).

```
5 REM*** PRG.#32-SORT DEMO ***
10 POKE53280,13:POKE53281,13:PRINTCHR$(3
0);CHR$(147)
20 FORJ=1TO6:B$=B$+CHR$(32):NEXT
30 N=20:DIMS(N):FORJ=1TON:S(J)=INT(RND(1
)*1000):PRINTTAB(3)S(J):NEXT
40 GOSUB1000
50 END
1000 FORJ=1TON:GOSUB4000:S=0:FORK=1TON-J
1005 GOSUB2000
```

```
1010  T=S(K):IFT>S(K+1)THENS(K)=S(K+1):S(
K+1)=T:S=1:GOSUB3000
1020  NEXTK:IFS=0THENJ=N:GOSUB6000
1025  GOSUB5000
1030  NEXTJ:RETURN
2000  PRINTCHR$(19):FORL=1TOK:PRINTCHR$(3
2):NEXT
2010  PRINTCHR$(145);CHR$(156);CHR$(246):
PRINTCHR$(246);CHR$(30)
2020  GOSUB9000:RETURN
3000  PRINTCHR$(145);
3010  FORL=3TO9
3020  PRINTTAB(L);CHR$(32);S(K);CHR$(145)
3030  GOSUB9000:NEXT
3040  PRINTCHR$(145);TAB(3)B$
3050  PRINTTAB(3);S(K+1);B$;CHR$(145);CHR
$(145)
3060  FORL=9TO3STEP-1
3070  PRINTTAB(L);S(K);CHR$(157);B$;CHR$(
145)
3080  GOSUB9000:NEXT
3090  RETURN
4000  PRINTCHR$(19);CHR$(144);"PASSO  #"J;
CHR$(30)
4010  RETURN
5000   :PRINTCHR$(19):FORL=1TO20:PRINTCHR$
(32):NEXT
5100  RETURN
6000  PRINTCHR$(19);TAB(10);CHR$(144)"FIN
E CAMBI:  SORT  COMPLETO"
6010  FORL=1TO18:PRINTCHR$(32):NEXT
9000  FORM=1TO5:NEXT:RETURN
```

Sid e Vic

............................Sid

............................Generatore di Caratteri

............................Sprite/screen

............................Sid

Il valore di un programmatore del chip SID del Commodore 64 non e' naturalmente calcolabile se tutto avviene nel silenzio piu' completo. La maggior parte degli effetti sonori contenuti nei programmi di questo libro consistono in semplici suoni, cosi' ho deciso di lasciarvi la possibilita' di valutare le mie capacita' musicali.

Il SID non puo' essere persuaso a produrre interessanti effetti sonori in un modo piu' semplice del seguente programma.
Viene visualizzata una lista dei principali parametri sonori e grazie all'uso dei due tasti-cursore potete posizionarvi sul parametro per inserire il valore desiderato (premete RETURN prima, cosi' come dopo aver digitato ogni numero, per "congelare" il cursore). Alla fine, quando avrete definito il suono da voi desiderato, premete S per ascoltarlo.

Una volta che S e' stato premuto, e' possibile alterare i parametri sonori mentre la musica sta suonando. Col cambio della VOCE, voi potete definire fino a tre suoni differenti, e sentirli uno alla volta o tutt'e tre contemporaneamente.

Per interrompere il suono di ogni voce, premete F (questo interrompera' il suono della voce visualizzata sullo schermo).

E' da notare che se non specificate un valore di SUSTAIN molto lungo, il suono finira' prima che facciate in tempo a premere F. Comunque la voce rimane attiva, e per ascoltare le vostre nuove composizioni vi bastera' premere F prima ancora di S.

Ho pre-programmato la prima voce con i DATA presenti nel programma dimostrativo del Manuale d'uso, nel capitolo 7. Consultate questo paragrafo (e l'appendice P) per avere ulteriori informazioni sulla formazione degli altri parametri, cercando magari di abbinarli a dei valori casuali per vedere (o meglio sentire) cosa succede.

```
5 REM*** PRG.#33-SID ***
10 SID=54272
20 DIMS(3,5):GOSUB800
30 PRINTCHR$(147);CHR$(5)
40 PRINT"■VOCE #  (1-3)",V
50 PRINT"■VOLUME",,S(V,0)
60 PRINT"■ATTACK/DECAY",S(V,1)
70 PRINT"■SUSTAIN/RELEASE",S(V,2)
80 PRINT"■NOTE (ALTA FREQ)",S(V,3)
90 PRINT"■NOTE (BASSA FREQ)",S(V,4)
100 PRINT"■WAVEFORM",,S(V,5)
110 PRINT"■■USA 'CRSR UP/DOWN' PER SCEGL
IERE"
120 PRINT"■POI 'RETURN' PER IMMETTERE IL
 VALORE"
130 PRINT"■USA 'S' PER SENTIRE OGNI VOCE
"
140 PRINT"■E 'F' PER FINIRE."
200 C=1
300 GETI$
310 IFI$="S"THENGOSUB1000
320 IFI$="F"THENGOSUB2000
330 IFI$=CHR$(13)THEN500
340 NC=C-(I$=CHR$(17))+(I$=CHR$(145)):IF
NC<1ORNC>7THENNC=C
350 PRINT"■■":FORJ=1TOC:PRINT"■":NEXT
360 PRINT"]]"TAB(18);CHR$(32);CHR$(32)
370 C=NC
380 PRINT"■■":FORJ=1TONC:PRINT"■":NEXT
390 PRINT"]]"TAB(18);CHR$(62);
400 GOTO300
500 INPUTI$
510 I=INT(VAL(I$))
520 IFC=1AND(I<1ORI>3)THEN700
530 IFC=2AND(I<0ORI>15)THEN700
540 IFC>2ANDI>255THEN700
600 IFC=1THENV=I:GOTO30
610 S(V,C-2)=I:GOTO300
700 PRINT"]"TAB(19);:GOTO500
800 V=1
810 S(V,0)=15
820 S(V,1)=190
830 S(V,2)=248
840 S(V,3)=17
850 S(V,4)=37
860 S(V,5)=17
870 RETURN
1000 POKESID+24,S(V,0)
1010 P=SID+7*(V-1)
1020 POKEP+5,S(V,1)
1030 POKEP+6,S(V,2)
1040 POKEP+1,S(V,3)
1050 POKEP,S(V,4)
1060 POKEP+4,S(V,5)
1070 RETURN
2000 P=SID+7*(V-1)
2010 POKEP+4,0
2020 POKEP+5,0
2030 POKEP+6,0
2040 RETURN
```

................Generatore di caratteri

Questo programma vi permette di creare dei vostri personali caratteri da usare al posto dei caratteri del Commodore 64.
Selezionate quale carattere desiderate sostituire e ridefinire, POKEando il valore da 0 a 255. Per un uso serio, e' preferibile selezionare uno dei caratteri grafici poco usati (quelli inversi, ad es., dal codice 192 al 255), ma ridefinire l'alfabeto (da 1 a 26) puo' essere anche divertente, rendendo ad esempio impossibile la lettura di un testo (ridefinire un carattere significa cambiare nella nuova forma ogni carattere presente sullo schermo).

Premete 0 se volete un punto del colore dello sfondo, e un 1 per vederlo col colore di stampa (che formera' una parte del carattere). A mano a mano che voi proseguite verranno visualizzati sia la griglia 8x8 del nuovo carattere in formazione, sia il modo in cui il vecchio carattere sta cambiando per assumere la sua nuova veste.

Una caratteristica aggiuntiva del programma e' quella che, una volta che risulterete soddisfatti del vostro nuovo carattere, una linea di DATA verra' creata e inserita nel programma (a partire dalla linea 10000), contenente il codice del carattere e gli otto numeri da POKEare per ottenere questo carattere al momento desiderato. Facendo una nota di queste istruzioni DATA, sarete in grado di utilizzare questo programmino includendolo in altri programmi piu' elaborati.

......................Note del programma

Questo programma, ovviamente, e' molto simile al CHR SET, comprese le linee 5-70 che sono praticamente identiche. L'unica differenza sta nella linea 40, dove solo i primi 256 caratteri vengono copiati dalla ROM alla RAM. Cio'

serve solo a risparmiare tempo, e significa che sono disponibili solo i caratteri del modo grafico (se premete i tasti+ il tasto Commodore o SHIFT non sarete in grado di ridefinire i caratteri, ne' di commutare i vari set di caratteri).

La linea 20 disabilita la tastiera mentre la "copiatura" sta avvenendo, e la linea 60 la ri-abilita. La linea 70 definisce quale set di caratteri (ROM o RAM) viene usato.

Trasferire il set di caratteri dalla ROM alla RAM significa che e' possibile non leggerlo soltanto, ma anche alterarlo tramite POKE con dei nuovi valori, ed e' esattamente quello che succede in questo programma. Non appena fate entrare il valore di ogni bit , la linea 275 crea la figura del vostro carattere sullo schermo, e la linea 280 conserva il valore totale del byte. Una volta che avrete fatto entrare gli otto bytes, il valore totale viene POKEato nell'appropriata posizione nel set dei caratteri alla linea 290. Questo processo e' ripetuto per gli altri sette bytes, fino a che l'intero carattere viene ridefinito.

Nel frattempo, l'array N memorizza gli otto bytes usati per definire il carattere in modo tale che, non appena decidete che e' accettabile, la linea di DATA venga creata, usando una routine non diversa da quella di Auto-line.

```
3 REM *** PRG.#34 - GENERAT.CARATTERI***
5 POKE53281,6:PRINTCHR$(147);CHR$(5);"PE
R FAVORE ATTENDERE ......"
10 POKE52,48:POKE56,48
20 POKE56334,PEEK(56334)AND254
30 POKE1,PEEK(1)AND251
40 FORJ=0TO2047:POKE12288+J,PEEK(53248+J
):NEXT
50 POKE1,PEEK(1)OR4
60 POKE56334,PEEK(56334)OR1
70 POKE53272,(PEEK(53272)AND240)+12
90 A=10000
100 PRINTCHR$(147)"CARATTERE DA RIDEFINI
RE <0-255>?"
110 INPUTC
120 PRINT"RIDEFINIRE CARATTERE #"C
130 D=54272:POKE1211,C:POKE1211+D,1
140 PRINT"USANDO INPUT BINARIO"
150 PRINT"0=BACKGROUND"
```

```
160 PRINT"1=FOREGROUND"
200 PRINTCHR$(17);CHR$(17);CHR$(17)
210 FORJ=0TO7
220 PRINT"ROW"J;TAB(15);
230 N(J)=0:FORK=0TO7
240 PRINTCHR$(156);CHR$(166);CHR$(5);
250 GETI$:IFI$<"0"ORI$>"1"THEN250
260 PRINTCHR$(157);I$;
270 X=13:IFI$="1"THENX=2
275 POKE1434+POS(1)+40*J,160:POKE55706+P
OS(1)+40*J,X
280 N(J)=N(J)+(VAL(I$))*2↑(7-K):NEXTK:PR
INT
290 POKE12288+8*C+J,N(J)
300 NEXTJ
310 PRINT"OK?"
320 GETI$
330 IFI$="N"THEN100
340 IFI$<>"Y"THEN320
500 PRINTCHR$(147):PRINT:PRINTA"DATA";C;
510 FORJ=0TO7:D$=STR$(N(J)):D$=RIGHT$(D$
,LEN(D$)-1)
520 PRINT",";D$;
530 NEXT:PRINT:PRINT"A="A+10":GOTO600"
540 POKE198,2:POKE631,13:POKE632,13
550 PRINTCHR$(19);:END
600 PRINTCHR$(147)"ANY MORE?"
610 GETI$
620 IFI$="Y"THEN100
630 IFI$<>"N"THEN610
640 LIST10000-
```

............................Sprite-schermo

Questo programma e' sostanzialmente simile a Cursore e Colore visto nel Capitolo 1, ma questa volta l'oggetto da muovere lungo lo schermo non e' un caraattere ma uno sprite. Lo schermo mostra inoltre le coordinate x e y della posizione dello sprite, e il "flag" x1 (il bit piu' significativo della coordinata x) tradotto allo stesso modo del normale carattere-video.

Lo sprite e' inizialmente posizionato interamente sullo schermo, nell'angolo in alto a sinistra, mostrando le seguenti coordinate:

SPRITE			SCHERMO	
X	Y	X1	X	Y
24	50	0	0	0

Esso puo' essere mosso in altre posizioni, fino alla parziale o totale uscita dallo schermo, tramite l'uso dei tasti di controllo-cursore. Se muovete lo sprite verso la destra, la coordinata X aumentera' fino ad un massimo di 255, dopo che il flag X1 sara' cambiato in 1. Le coordinate dello schermo vi riferiscono in ogni momento l'equivalente posizione tramite i valori visualizzati in alto sullo schermo.

Poiche' la risoluzione dello "Sprite-schermo" e' 64 volte piu' grande del carattere-schermo, le coordinate-video saranno incrementate da uno step di 0.125 per ogni incremento di coordinata sprite di 1.

Solo quando le coordinate x e y dello schermo saranno formate da valori interi in alto a sinistra dello schermo, corrisponderanno esattamente ad una posizione normale del carattere.

Il valore elativo delle coordinate e' utile specialmente quando si tratta di alterare la posizione per evitare collisioni con caratteri presenti sullo schermo (per un esempio, guardate piu' avanti il programma "Skyline").

```
5 REM*** PRG.#35- SPRITE SCREEN ***
10 FORJ=0TO62:POKE832+J,255:NEXT:REM DEF
INE SPRITE
20 POKE2040,832/64:REM LOCATE SPRITE DAT
A
30 X=24:Y=50:X1=0:POKE53287,13:POKE53275
,1:POKE53269,1:REM COLOUR AND SET SPRITE
40 PRINTCHR$(147);CHR$(5)TAB(1)"SPRITE",
TAB(19)"SCHERMO"
50 PRINTTAB(1)"X"TAB(6)"Y"TAB(13)"X1"TAB
(19)"X"TAB(27)"Y"TAB(35)"PEEK"
60 FORJ=1TO39:B$=B$+CHR$(32):NEXT:GOSUB5
00
70 POKE53248,X
80 POKE53249,Y
90 POKE53264,X1
100 GETI$:IFI$=""THEN100
110 IFI$=CHR$(17)ANDY<255THENY=Y+1
120 IFI$=CHR$(145)ANDY>0THENY=Y-1
130 IFI$=CHR$(157)THENGOSUB200
140 IFI$=CHR$(29)THENGOSUB250
150 GOSUB500:GOTO70
200 IFX>0THENX=X-1:RETURN
210 IFX=0ANDX1=1THENX=255:X1=0
220 RETURN
250 IFX<255THENX=X+1:RETURN
260 IFX=255ANDX1=0THENX=0:X1=1
```

```
270 RETURN
500 PRINTCHR$(19);CHR$(17):PRINTB$:PRINT
CHR$(145)X;TAB(5);Y;TAB(12);X1;TAB(18);
510 X2=X+256*X1
520 SX=(X2-24)/8
530 SY=(Y-50)/8
540 IFSX<0ORSX>=40ORSY<0ORSY>=25THENL=0:
GOTO600
550 L=1024+INT(SX)+40*INT(SY)
600 PRINTSX;TAB(26);SY;TAB(34);L
610 RETURN
```

Interludio

............................Censura

Grazie a questo piccolo programma vi potrete divertire a censurare i testi prodotti dal vostro Commodore 64.

Il testo prodotto e' composto da lettere casuali; il programma non e' in grado (per il momento) di scrivere oscenita', ma credo che presto i piu' perversi tra di voi avranno apportato le necessarie modifiche affinche' cio' avvenga.

Decidete quale lettera eliminare e premete il tasto corrispondente. Quando sarete soddisfatti al punto da poter far leggere perfino a vostra nonna quello che appare sullo schermo, premete il tasto 0, e guardate il testo finale, ripulito dal computer di tutte le tracce della vostra censura. (Questo momento finale del programma e' certamente da guardare attentamente).

```
5 REM*** PRG.#36-CENSURA ***
10 POKE53280,1:POKE53281,1:PRINTCHR$(147
);CHR$(144);
20 FORJ=1TO28:B$=B$+CHR$(32):NEXT
30 FORJ=0TO839
40 P=INT(RND(1)*26)+65
50 IFRND(1)>.7ANDLP<>32THENP=32
60 PRINTCHR$(P);
70 LP=P
80 NEXT:PRINT
90 PRINTCHR$(28)TAB(6)"QUALE LETTERA VUO
I CENSURARE?"
100 GETI$
110 IFI$<"0"ORI$>"Z"THEN100
120 IFI$="0"THEN220
130 POKE53280,INT(RND(1)*14)+2
140 PRINTCHR$(145);:PRINTTAB(6);B$
150 FORJ=0TO839
160 IFPEEK(1024+J)<>ASC(I$)-64THEN180
170 POKEJ+1024,160:POKEJ+55296,6
180 NEXT
190 POKE53280,1
200 PRINTCHR$(145);
210 GOTO90
220 PRINTCHR$(145);
230 PRINTCHR$(30)TAB(6)LEFT$(B$,4)"MESSA
GGIO ORA APPROVATO"LEFT$(B$,4)
240 LP=0:POKE53280,5
250 K=1024:FORJ=KTOK+839
260 P=PEEK(J)
270 POKEJ,32:POKEJ+54272,0
280 IFP=160THENP=LP:GOTO300
290 IF(LP<>32ORP<>32)THENPOKEK,P:K=K+1
300 LP=P:NEXT
310 PRINTCHR$(145);
320 PRINTTAB(6)LEFT$(B$,5)"NON E' MEGLIO
?"LEFT$(B$,5)
330 POKE53280,1
340 GETI$:IFI$=""THEN340
```

....................Inchiostro invisibile

In questo programma, la tastiera del Commodore 64 lavora come una qualsiasi macchina da scrivere, ma se qualcuno

entra nella stanza mentre state scrivendo, basta premere F1 ed il testo diventera' immediatamente invisibile.
Una volta che gli intrusi saranno usciti, premendo F7 il testo ricomparira'
(Non per niente il sotto-titolo di questo capitolo si chiama "la sezione strampalata")

```
5 REM***PRG.#37-INCHIOSTRO INVISIBILE***
10 POKE53280,14:POKE53281,1
20 PRINTCHR$(147);CHR$(31);CHR$(14);
30 PRINT"♥CRIVI PURE : USA 'F1' PER NAS
CONDERE":REM HEART IS CAPITAL 'S'
40 PRINTTAB(19)"'F7' PER SVELARE"
50 GETI$:IFI$=""THEN50
60 IFI$=CHR$(133)THENPOKE53281,6
70 IFI$=CHR$(136)THENPOKE53281,1
80 PRINTI$;
90 GOTO50
```

....................List invisibile

Sebbene sembri stupido come il precedente programma, questo accorgimento puo' avere utili ed interessanti applicazioni. Con l'uso appropriato dei caratteri di controllo in una linea di programma, e' possibile impedire che venga listata.

Se battete la linea 10 seguendo le istruzioni scritte nelle altre sei linee del programma vi troverete alla fine con due linee a prima vista simili; ma se chiederete il LIST, la prima linea non comparira'.

Come si puo usare? Naturalmente un programma interamente costituito da istruzioni REM e' inutilizzabile, ma la stessa tecnica per nascondere delle linee di programma o una parte di esse puo' essere usata in un programma che desiderate usare solo voi. Infatti e' possibile nascondere una parola d'ordine (password) che tramite un INPUT controlli segretamente se una persona e' autorizzata o no per proseguire.

.... Esagrammi

Non ho idea del perche' cosi' tanti programmatori di computer siano ossessionati nell'usare il loro sofisticato equipaggiamento elettronico per produrre degli esagrammi cinesi. Forse ha qualcosa a che fare col fatto che cosi' tanti computer vengono costruiti in California?

```
5 REM*** PRG.# 38 - LIST INVISIBILE ***
10 REM QUI SONO 30 CARATTERI :"■■■■■■■■■
■■■■■■■■■■■■■■■■■■■■■
20 REM QUI SONO 30 CARATTERI:
30 REM SEGUITI DA ""
40 REM SEGUITI DA CURSORE SINISTRA
50 REM SEGUITI DA 31 INSERTI
60 REM SEGUITI DA 31 'DELETE'
70 REM SEGUITI DA UNO SPAZIO PER CANCEL
LARE IL FINALE"
```

.......................Crittogrammi

Questo programma utilizza la funzione RND del Commodore per creare dei codici indecifrabili. Il concetto dell'uso della funzione RND con un argomento negativo e' stato gia' discusso nel primo capitolo (il "Gioco dei dadi"), dove vi era stato suggerito come ottenere dei numeri casuali difficilmente prevedibili grazie all'uso di RND(-TI). Comunque, conoscendo un argomento negativo, il risultato di ogni sequenza di numeri casuali puo' essere riprodotta esattamente sia dopo qualche tempo, sia da qualsiasi persona che sappia usare il computer.

Teoricamente, occorrerebbe un codice ideale che sostituisse nel testo originale le lettere con delle lettere casuali, ed e' precisamente quello che fa questo programma. Ma poiche' usiamo un argomento negativo che solo noi conosciamo, il messaggio in codice puo' essere facilmente convertito al testo originale semplicemente applicando al contrario lo stesso valore RANDOM.

Nota Bene: quando il programma richiede un valore (linea 20), digitate un numero POSITIVO, che sara' reso negativo alla linea 30.

```
5 REM*** PRG.#39-CRITTOGRAMMI ***
10 POKE53280,5:POKE53281,13
20 PRINTCHR$(147)
30 FORJ=1TO14:A$=A$+CHR$(32):NEXT
40 FORJ=1TO5:PRINT:NEXT
50 FORJ=1TO11:PRINTTAB(13);CHR$(158);CHR
$(18)A$:NEXT
60 PRINT:PRINTTAB(4);CHR$(144)"PREMI SPA
ZIO PER L'ESAGRAMMA"
70 GOSUB500
80 X=RND(-TI)
90 REM 55549
100 FORJ=0TO5
110 X=RND(1)
120 IFX>.5THEN200
130 FORK=0TO1
140 POKE55555+K+80*J,13
150 NEXT
200 FORK=1TO14
210 POKE55588+K+80*J,13
220 NEXTK,J
300 PRINTCHR$(145);TAB(20)" CONTINUARE."
310 GOSUB500
320 RUN
500 GETI$:IFI$<>CHR$(32)THEN500
510 RETURN
```

.......................Invertire

Se avete gia' digitato i programmi "CHR Set" e "Generatore di caratteri" contenuti in questo libro, vi dovrebbe risultare abbastanza chiaro come lavora il programma "Invertire". Esso e' in grado di provocare sorprendenti effetti, riuscendo a provocare della costernazione in chi lo vede girare sul proprio computer o su un modello in un negozio.

Come "CHR Set", l'intero set di caratteri viene copiato dalla CHRGEN ROM alla RAM, ma con una piccola differenza. Ogni caratterre e' memorizzato al contrario.

Il programma termina con l'istruzione LIST alla linea 80, ma se voi digitate GOTO 200 potrete vedere un altro effetto curioso: infatti i caratteri torneranno nella normale posizione, "ribaltandosi" il set dei caratteri correnti (i valori bassi dell'indirizzo 53272).

```
3 REM *** PRG.# 40-INVERT ***
5 PRINTCHR$(147)"PER FAVORE ATTENDERE...
."
10 POKE52,48:POKE56,48
20 POKE56334,PEEK(56334)AND254
30 POKE1,PEEK(1)AND251
40 GOSUB100:REM CREATE NUOVO SET DI CARA
TTERI
50 POKE1,PEEK(1)OR4
60 POKE56334,PEEK(56334)OR1
70 POKE53272,(PEEK(53272)AND240)+12
80 LIST
100 FORJ=0TO255
110 FORK=0TO7
120 POKE12288+J*8+7-K,PEEK(53248+J*8+K)
130 NEXTK,J
140 RETURN
200 FORJ=5TO12STEP7:POKE53272,(PEEK(5327
2)AND240)+J:FORK=1TO500:NEXTK,J:GOTO200
```

Giocaci SAM

....................Cattura il mio cuore

Questo gioco e' talmente semplice da sembrare banale, e sotto qualche aspetto e' simile alla vita stessa. Dovete rimanere vivi il piu' a lungo possibile, pena la fine del gioco, e catturare quanti piu' cuori potete.

Il vostro segnalino e' un diamante giallo, e puo' essere mosso grazie ai due tasti-cursore (il cursore vertic. muove a sinistra, il cursore orizz. muove a destra). Voi dovete tentare di catturare senza farvi colpire dalle frecce che li circondano. Non appena sarete colpiti da una

freccia, il gioco sara' finitoe verra' visualizzato il vostro punteggio, insieme al tempo in cui siete riusciti a sopravvivere.

..................Note del programma

Il programma sfrutta una caratteristica del Commodore 64 che non e' normalmente considerata vantaggiosa, cioe' la brutta abitudine di scrollare l'intero video quando viene digitata l'ultima linea dello schermo. In tal modo non appena qualcosa viene scritto sull'ultima linea, questa non e' gia' piu' l'ultima linea.

La linea 110 PRINTa le freccie, i cuori, le stringhe delle freccie (come definite dai caratteri delle istruzioni DATA - alla linea 1000) seguendo una casuale TAB lungo la linea finale causando l'immediato scrolling e provocando il movimento dei caratteri. Il carattere diamante, comunque, viene fermato grazie alle linee 120 e 50.

L'effetto finale mostra un veloce movimento delle frecce e dei cuori verso il bordo alto dello schermo, mentre una nuova stringa viene sempre visualizzata sull'ultima linea visibile del video.

```
5 REM*** PRG.#41-CATTURA IL MIO CUORE***
10 POKE53280,6:POKE53281,14:PRINTCHR$(14
7)
20 P=1564:D=54272
30 READJ:IFJ<>999THENR$=R$+CHR$(J):GOTO3
0
40 FORJ=1TO24:PRINT:NEXT:TI$="000000"
50 POKEP,90:POKEP+D,7
60 G=PEEK(P+40)
70 IFG=83THENS=S+1
80 IFG=30THEN200
90 GETI$
100 Q=P+((I$=CHR$(29))*(P<1582))-((I$=CH
R$(17))*(P>1545))
110 PRINTTAB(INT(RND(1)*34)+1);R$
120 POKEP+D-40,14
130 P=Q
140 GOTO50
200 T$=TI$
210 PRINTCHR$(5)"██HAI CATTURATO"S"CUORE
";:IFS<>1THENPRINT"█I";
```

```
220 T=VAL(MID$(T$,3,2))*60+VAL(RIGHT$(T$
,2))
230 PRINT:PRINT"E SEI SOPRAVVISSUTO PER"
T"SECONDO";:IFT<>1THENPRINT"▮▮I"
240 PRINT:PRINT:PRINT"▮▮PREMI LO SPAZIO
PER GIOCARE ANCORA"
250 GETI$:IFI$<>CHR$(32)THEN250
260 RUN
1000 DATA144,94,32,28,115,144,32,94,999
```

..........................Reversi

L'antico gioco di Reversi e' stato fedelmente riprodotto in questo programma. Nella spiacevole eventualita' che voi non abbiate mai giocato a questo simpatico passatempo, le istruzioni (di cui vi propongo un sommario piu' sotto) vi verranno proposte all'interno del programma (linee 1000-10360). Il programma ha cinque diversi livelli di gioco, prevedendo l'opzione di giocare contro il computer, in due giocatori (l'uno contro l'altro), di salvare il gioco iniziato su disco (o anche su cassetta, basta una piccola modifica) per finirlo in un altro momento.

L'opzione "ripresa" permette di ripetere un gioco nel modo esatto, dandovi il completo controllo ad ogni punto del programma.

Il computer gioca in un modo abbastanza lento, e anche se gioca da esperto, e' lontano dalla perfezione. In questo modo, a meno che voi non siate dei campioni di Reversi, troverete nel vostro Commodore un degno avvversario.

......................Istruzioni

Il gioco si svolge su una scacchiera 8x8.

Una "cattura" viene effettuata chiudendo tra due propri pezzi uno o piu' pezzi dell'avversario lungo un asse che puo' essere orizzontale, verticale o diagonale; le pedine catturate vengono girate ed assumono il colore di chi le ha "mangiate".

Una singola mossa puo' provocare piu' di una cattura. Ad ogni mossa deve corrispondere almeno una cattura, altrimenti la mossa non e' valida.

Nel caso che un giocatore non possa effettuare catture, la mano passa all'avversario.

Il gioco e' finito quando:

Un giocatore non ha piu' pezzi sulla scacchiera.

Nessun giocatore puo' effettuare mosse valide o

Tutti i riquadri della scacchiera sono stati occupati.

In tutti i casi, il vincitore sara' il giocatore che possiedera' piu' pezzi alla fine del gioco.

E' da notare che, poiche' e' impossibile soddisfare l'istruzione riguardante mosse non permesse partendo da una scacchiera vuota, i primi quattro pezzi vengono predisposti sulla griglia prima dell'inizio del gioco.

```
0 REM*** PRG.# 42-REVERSI ***
5 POKE53280,13:POKE53281,1:PRINTCHR$(30)
10 PRINT"■";TAB(16);"■■■■■■■■■■REVERSI■■■■
■■■■■■■■■■■BY   V&H
15 N=1320:GOSUB100
20 CH=1024:D=54272
25 FORJ=CHTO1063:POKEJ,83:POKEJ+D,2:NEXT
30 GOSUB100
35 FORJ=1984TO2023:POKEJ,83:POKEJ+D,2:NE
XT
40 GOSUB100
45 FORJ=1064TO2023STEP40:POKEJ-1,83:POKE
J,83:POKEJ+D,2:POKEJ+D-1,2:NEXT
50 GOSUB100
55 FORJ=1TO100:IFJ/10=INT(J/10)THENGOSUB
100
60 NEXT:GOTO150
100 FORK=0TO6:POKEN+K,PEEK(N+K)-(256*(PE
EK(N+K)<128))-128:NEXTK:RETURN
110 CP$=CHR$(31)+CHR$(215)+CHR$(30)
120 YP$=CHR$(28)+CHR$(209)+CHR$(30)
130 RETURN
150 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■VUOI LE
 ISTRUZIONI ?"
155 A$="":GETA$:IFA$=""THEN155
156 IFA$="S"THEN159
157 IFA$<>"N"THEN155
158 IFA$="N"THEN160
159 GOSUB110:GOSUB10000
```

```
160 IFLEFT$(A$,1)="S"THENGOSUB110:GOSUB1
0000
165 CLR:DIMG(60),PG(60),SPG(60):C1=1147:
C2=1161:C3=1707:C4=1721:D=54272
170 GOSUB110
180 PRINTCHR$(30)
185 PRINT"■■       PREGO SCEGLI IL TIPO DI
GIOCO"
190 PRINT"────────────────────────────────────
195 PRINT"■■(GIOCHI CONTRO IL C=64:   LIV
ELLO 1-5)
200 PRINT"■■1: PRINCIPIANTE
210 PRINT"■2: MEDIO
220 PRINT"■3: GIOCO AGGRESSIVO
230 PRINT"■4: GIOCO TATTICO
240 PRINT"■5: ESPERTO
250 PRINT"■6: DUE GIOCATORI
260 PRINT"■7: CARICA GIOCO PRECED. DA DI
SCO
270 LV$="":GETLV$:IFLV$=""THEN270
280 LV=VAL(LV$):IFLV<1ORLV>7THEN270
290 IFLV=7THEN15000
300 IFLV=6THEN500
310 PRINT"■■■■■■■VUOI GIOCARE PER PRIMO
?
320 PRINT"■■■■■■S= SI'
330 PRINT"■N= NO
340 PRINT"■T= LASCI CHE DECIDA IL COMPUT
ER
350 FP$="":GETFP$:IFFP$=""THEN350
360 IFFP$="T"THENGOSUB9000
370 IFFP$="S"ORFP$="N"THEN500
380 GOTO350
500 REM ** CAMPO **
510 PRINT"■ ■REVERSI■  BY V&H          ";
:IFLV=6THENPRINT:PRINT:PRINT:GOTO530
520 PRINTYP$"= TU ■":PRINTTAB(25);CP$"=
COMPUTER"
530 FORJ=1TO8:PRINTCHR$(156);
540 PRINT9-J;
545 PRINTCHR$(144);:FORK=1TO8:PRINTCHR$(
166);CHR$(32);:NEXT:PRINT"■"
550 NEXT
560 PRINTTAB(3);CHR$(156);
570 FORJ=1TO8:PRINT"■■";J;:NEXT:PRINTCHR$
(30)
580 FORJ=1TO500:NEXT:PRINT
590 POKE1393,81:POKE1475,81:POKE1395,87:
POKE1473,87
595 POKE55665,2:POKE55747,2:POKE55667,6:
POKE55745,6
600 GOSUB2000
605 IFA=2THEN30050
610 IFFP$="S"THENB$=YP$:GOSUB3000
615 IFZ=1THEN20000
620 IFLV<>6THEN650
625 REM
630 NM(2)=0:FP=1:B$=YP$:GOSUB3000:IFZ=1T
HEN20000
640 NM(0)=0:FP=-FP:B$=CP$:PRINT"■":GOSUB
3000:IFZ=1THEN20000
645 GOTO625
650 IFFP$="N"THENF=1
660 FP$="S":GOSUB6000
670 IFZ=1THEN20000
700 GOTO610
```

```
1000  REM ** TEST PER FINIRE **
1010  Z=0
1020  IFNPM=1ANDNCM=1THENZ=1
1030  IFCO=0ORCG=0THENZ=1
1040  IFLV=6ANDNM(0)=1ANDNM(2)=1THENZ=1
1050  IFCG+CO=64THENZ=1
1060  RETURN
2000  CG=0:CO=0
2010  FORM=C1TOC3STEP80
2020  FORK=MTOM+14STEP2
2030  IFPEEK(K)=81THENCG=CG+1
2040  IFPEEK(K)=87THENCO=CO+1
2050  NEXTK:NEXTM
2060  PRINT"▮▮▮▮▮▮▮▮▮▮";TAB(25);YP$"=
▮▮▮▮▮▮▮";CG
2070  PRINT"▮▮▮";TAB(25);CP$"=                ▮▮▮▮▮▮▮
";CO
2080  PRINT"▮▮▮";TAB(20);"TOTALE=            ▮▮▮▮
▮▮▮▮";CO+CG
2090  PRINT"▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮"
2100  RETURN
3000  REM
3010  PG=0:NPM=0
3020  PRINT"TUA MOSSA (";B$;:INPUT">      *
      ▮▮▮▮▮▮▮▮▮▮▮▮";A$
3030  IFA$="00"THENGOSUB3900:IFNPM=1THEN3
130
3040  IFA$="*"THENPRINT"▮▮":GOTO3020
3050  IFVAL(A$)<11ORVAL(A$)>88THENA$="*":
GOTO3030
3060  IFRIGHT$(A$,1)="9"ORRIGHT$(A$,1)="0
"THENA$="*":GOTO3030
3070  L=VAL(LEFT$(A$,1)):R=VAL(RIGHT$(A$,
1))
3080  LL=1785-(80*L)+(2*R)
3090  IFPEEK(LL)<>102THENA$="*":GOTO3030
3100  G=81:O=87:IFFP=-1THENG=87:O=81
3110  GOSUB5000
3120  IFA$="*"THEN3040
3130  GOSUB2000:IFNPM=1THEN3150
3140  G(CG+CO-4)=LL
3150  GOSUB1000:RETURN
3900  G=81:O=87:IFFP=-1THENG=87:O=81
4000  REM
4003  FORJ=1TO60:PG(J)=0:SPG(J)=0:NEXTJ
4005  PG=1
4010  FORM=C1TOC3STEP80
4020  FORN=MTOM+14STEP2
4030  IFPEEK(N)<>102THEN4070
4040  LL=N
4050  GOSUB5000
4060  IFPG<>1THENA$="*":GOTO4090
4070  NEXTN:NEXTM
4080  NPM=1:IFLV=6THENNM(FP+1)=1
4090  PG=0
4100  RETURN
5000  REM
5010  FORJ=0TO7:L(J)=0:NEXTJ
5020  L(0)=LL:Y=0
5030  FORJ=1TO8
5040  X=J=1AND-80ORJ=2AND-78ORJ=3AND2ORJ=
4AND82ORJ=5AND80ORJ=6AND78ORJ=7AND-2
5050  X=XORJ=8AND-82
5060  FORK=1TO7
5070  L(K)=L(K-1)+X
5080  IFL(K)<C1ORL(K)>C4THENL(K)=1024
5100  NEXTK
```

```
5110  IFPEEK(L(1))<>0THEN5200
5120  FORK=2TO7
5130  IFPEEK(L(K))=0THEN5150
5135  IFPEEK(L(K))=102ORPEEK(L(K))=32THEN
5200
5140  IFPEEK(L(K))=GTHEN5160
5150  NEXTK:GOTO5200
5160  IFPG=0THENGOSUB5500:GOTO5200
5170  PG(PG)=LL
5180  SPG(PG)=K+SPG(PG)-1
5190  IFLV<>5THEN5200
5195  IFJ/2=INT(J/2)THENSPG(PG)=SPG(PG)+2
5197  IF(L(K)=C1ORL(K)=C2ORL(K)=C3ORL(K)=
C4)THENSPG(PG)=SPG(PG)+25
5200  NEXTJ
5205  IFSPG(PG)<>0ANDPG<>0THENPG=PG+1
5210  IFY=0THENA$="*"
5220  RETURN
5500  Y=1:FORM=0TOK:POKEL(M),G
5510  POKEL(M)+D,(G-78)/1.5
5520  NEXT:RETURN
6000  REM
6005  NCM=0
6010  FORJ=1TO60:PG(J)=0:SPG(J)=0:NEXTJ
6020  PG=1:G=87:O=81
6025  Q=C1:QQ=C3:IF(CO+CG)<8THENQ=Q+80:QQ
=QQ-80
6026  IF(CO+CG)<6THENQ=Q+80:QQ=QQ-80
6030  FORM=QTOQQSTEP80
6040  FORN=MTOM+14STEP2
6050  IFPEEK(N)<>102THEN6100
6060  LL=N
6070  GOSUB5000
6080  IFPG=2ANDLV=1THENPG(0)=PG(1):GOTO62
00
6100  NEXTN:NEXTM
6110  IFPG=1THEN6300
6120  IFLV=2THENPG(0)=PG(INT(RND(1)*(PG-1
))+1):GOTO6200
6125  IF((CO+CG)>6)ANDLV>3THENGOSUB7000
6130  NR=0:SPG(0)=-100:FORK=1TOPG-1
6135  IFPG(K)=C1ORPG(K)=C2ORPG(K)=C3ORPG(
K)=C4THENSPG(K)=SPG(K)+40
6140  IFSPG(K)=SPG(0)THENNR=NR+1
6150  IFSPG(K)>SPG(0)THENSPG(0)=SPG(K):PG
(0)=PG(K):NR=1
6160  NEXTK:IFNR=1THEN6200
6170  RN=INT(RND(1)*NR)+1
6180  FORK=1TOPG:IFSPG(K)<>SPG(0)THEN6190
6185  RN=RN-1:IFRN=0THEN6195
6190  NEXTK
6195  PG(0)=PG(K)
6200  LL=PG(0)
6210  PG=0:GOSUB5000
6215  CM=(1785-LL)/80:M1=INT(CM+.5):M2=IN
T((M1-CM)*40+.5):M1=(M1*10)+M2
6220  PRINT"▓▓MIA MOSSA ("CP$")   ";M1
6230  GOSUB2000
6240  G(CO+CG-4)=LL
6250  GOTO6310
6300  PRINT"▓▓NON HO MOSSE                ":
GOSUB2000:NCM=1
6310  GOSUB1000:RETURN
7000  REM ** LEVEL 4&5 **
7010  FORK=1TOPG-1
7020  S=PG(K)
7030  IF(S=1229ORS=1227ORS=1239)ANDPEEK(C
```

```
1)<>87THENSPG(K)=SPG(K)-20
7035  IF(S=1239ORS=1241ORS=1159)ANDPEEK(C
2)<>87THENSPG(K)=SPG(K)-20
7040  IF(S=1629ORS=1627ORS=1709)ANDPEEK(C
3)<>87THENSPG(K)=SPG(K)-20
7045  IF(S=1639ORS=1641ORS=1719)ANDPEEK(C
4)<>87THENSPG(K)=SPG(K)-20
7060  IFLV=4THEN8100
7800  REM
7810  IFS<1162ORS>1706THENSPG(K)=SPG(K)+1
0
7820  IF(S-1147)/80=INT((S-1147)/80)THENS
PG(K)=SPG(K)+10
7830  IF(S-1161)/80=INT((S-1161)/80)THENS
PG(K)=SPG(K)+10
7880  IFPEEK(S-2)=87THENSPG(K)=SPG(K)+1
7890  IFPEEK(S+2)=87THENSPG(K)=SPG(K)+1
7900  IFPEEK(S+80)=87THENSPG(K)=SPG(K)+1
7910  IFPEEK(S-80)=87THENSPG(K)=SPG(K)+1
7920  IFPEEK(S-78)=87THENSPG(K)=SPG(K)+1
7930  IFPEEK(S+78)=87THENSPG(K)=SPG(K)+1
7940  IFPEEK(S+82)=87THENSPG(K)=SPG(K)+1
7950  IFPEEK(S-82)=87THENSPG(K)=SPG(K)+1
8100  NEXTK
8900  RETURN
9000  PRINT"TESTA PARTI TU : CROCE PA
RTO IO
9005  PS=1024
9010  RESTORE:FORJ=1TO25
9020  READSP:POKESP+D,7
9030  POKEPS,32:POKESP,100:POKESP,78:POKE
SP,84:POKESP,77:POKESP,100
9035  FORK=1TO90:NEXT
9040  PS=SP
9050  NEXT
9060  FP=INT(RND(1)*2)+1
9070  IFFP=1THENFP$="S":FP=8:PRINT"PA
RTI TU PER PRIMO
9080  IFFP=2THENFP$="N":FP=20:PRINT"P
ARTO IO PER PRIMO
9090  POKESP,FP
9100  FORJ=1TO3000:NEXT
9110  RESTORE:RETURN
9200  DATA1997,1957,1918,1878,1838,1798,1
759,1719,1679,1640,1600
9210  DATA1560,1521,1562,1602,1642,1683,1
723,1763,1804,1844,1884
9220  DATA1925,1965,2005
10000 REM ** REGOLE **
10010 PRINTCHR$(144)
10020 PRINT"                    REGOLE DI GIOCO
10030 PRINT"
10040 PRINT"IL GIOCO E' SIMILE AL FAM
OSO REVERSI,"
10050 PRINT"CON LE SEGUENTI ECCEZIONI:"
10070 PRINT"1:QUANDO GIOCHI CONTRO IL
COMPUTER       "
10075 PRINT"  (LIVELLI 1-5), TU SEI SEM
PRE ";YP$;CHR$(144)
10080 PRINT"  IL COMPUTER E' SEMPRE ";C
P$;CHR$(144)
10085 PRINT"  (PUOI SCEGLIERE COMUNQUE
DI PARTIRE    "
10088 PRINT"  PRIMO O SECONDO.)
      "
10090 GOSUB12000
10100 PRINT"2: QUANDO GIOCA IL SECOND
```

```
O GIOCATORE       "
10110 PRINT"     <LIVELLO 6>, IL GIOCATOR
E CHE    "YP$;CHR$(144)
10120 PRINT"     SCEGLIE PARTE PER PRIMO.
     "
10130 PRINT"3:  FATE LA VOSTRA MOSSA DIG
ITANDO 2"  "
10140 PRINT"     NUMERI, COORDINATA ORIZZ
ONTALE       "
10150 PRINT"     SEGUITA DALLA COORDINATA
 VERTICALE "
10160 PRINT"     LE MOSSE ILLEGALI SARANN
O RESPINTE."
10170 PRINT"4:  SE NON POTETE PROSEGUIRE
'00'.(CIO'  "
10180 PRINT"     SARA' RIFIUTATO SE INVEC
E SARETE     "
10190 PRINT"     IN GRADO DI MUOVERE)
   "
10195 GOSUB12000
10200 PRINT"5:  PUOI RIGIOCARE SUBITO,
 OPPURE      "
10210 PRINT"     SALVARE IL GIOCO SU DISC
O PER TERMI-"
10220 PRINT"     NARE IN SEGUITO. INTERRO
MPI IL GIOCO"
10230 PRINT"     SE VUOI FARLO, PRIMA DEL
LA MOSSA     "
10240 PRINT"     DEL COMPUTER, SE GIOCHI
AI LIVELLI   "
10250 PRINT"     1 - 5 .
             "
10350 PRINT"6:  ATTENDI SEMPRE IL CURSOR
E INTERMIT-"
10360 PRINT"       TENTE PRIMA DI FARE LE
 TUE MOSSE"
10370 PRINT"PREMI SPAZIO PER GIOC. O
R PER LE ISTR."
11300 POKE1684,42:POKE55956,5
11310 GOSUB12010
11320 IFA$="R"THEN10000
11330 PRINT""
11400 RETURN
12000 PRINT"
 PREMI SPAZIO PER CONTINUARE "
12005 FORJ=1TO20:IFPEEK(1684)=42THENPOKE
1684,170
12006 NEXT
12007 FORJ=1TO20:IFPEEK(1684)=170THENPOK
E1684,42
12008 NEXT
12010 A$="":GETA$:IFA$<>" "ANDA$<>"R"THE
N12005
12020 RETURN
15000 PRINT"                CARICO GIOCO
15010 PRINT"
15020 PRINT"PREGO SCRIVI IL NOME DEL G
IOCO CHE VA"
15030 PRINT"CARICATO."
15040 INPUT"       *    ";SN$
15050 IFSN$="*"THENSN$="REVERSI GAME"
15070 OPEN1,8,8,"0:"+SN$+"S,R"
15075 INPUT#1,SN$
15080 PRINT"CARICO   ";CHR$(34);SN$
;CHR$(34)
15090 INPUT#1,F
15100 INPUT#1,LV
```

```
15110 FORK=1TO60
15120 INPUT#1,G(K)
15130 NEXT
15140 CLOSE1
15145 A=2
15150 GOTO30000
20000 PRINT"
  ";:Z=0
20005 IFCO=CGTHENPRINT"A SORTE!
                    ":GOTO20070
20010 IFCO>CGANDLV=6THENB$=CP$
20020 IFCG>COANDLV=6THENB$=YP$
20030 IFCO>CGANDLV<>6THENB$="IO HO"
20040 IFCG>COANDLV<>6THENB$="TU HAI"
20050 PRINTB$;" VINTO";:IFLV=6THENPRINT"
S";
20060 PRINT" DA";ABS(CO-CG);"
  "
20070 PRINT"      PREMI SPAZIO PER CONTI
NUARE   "
20080 A$="":GETA$:IFA$=""THEN20080
20090 PRINT"                  OPZIONI
20100 PRINT"
20110 PRINT"1: NUOVO GIOCO
20120 PRINT"2: RIGIOCA L'ULTIMA PARTITA
"
20130 PRINT"3: SALVA IL GIOCO SU DISCO"
20135 PRINT"4: FINE"
20140 A$="":GETA$:IFA$=""THEN20140
20150 A=VAL(A$):IFA<1ORA>4THEN20140
20160 ONAGOTO165,30000,21000,31000
21000 PRINT"                SALVA GIOCO
21010 PRINT"
21020 PRINT"SE VUOI DARE A QUESTO GIOC
O UN TUO"
21030 PRINT"NOME, PER FAVORE DIGITALO A
DESSO."
21040 PRINT"ALTRIMENTI VERRA' SALVATO C
ON IL NOME"
21050 PRINT"DI ' REVERSI GAME ' ."
21055 PRINT"(BATTI X SE NON VUOI SALV
ARE IL GIOCO)"
21056 REM
21060 INPUT"     *":SN$
21070 IFSN$="*"THENSN$="REVERSI GAME"
21075 IFSN$="X"THEN20090
21080 OPEN1,8,8,"@0:"+SN$+",S,W"
21085 PRINT"SALVO ";CHR$(34);SN$;CH
R$(34)
21090 PRINT#1,SN$
21095 PRINT#1,F
21100 PRINT#1,LV
21110 FORK=1TO60
21120 PRINT#1,G(K)
21130 NEXT
21140 CLOSE1
21150 GOTO20090
30000 REM
30010 G=81:O=87
30020 IFF=1THENG=87:O=81
30030 GOTO500
30050 FORR=1TO60:CX=CO+CG
30060 IFG(R)=0THEN20000
30080 LL=G(R):PG=0
30090 GOSUB5000
30100 GOSUB2000
30102 IFCO+CG=CXTHENR=R-1
```

```
30105 O=G:G=G+6:IFG=93THENG=81
30110 PRINT"PREMI SPAZIO PER CONTINUARE▒
▒"
30120 IFLV=6ORG=81THENPRINT"]] Q PER INT
ERROMPERE"
30130 A$="":GETA$:IFA$=""THEN30130
30135 PRINT"TTT]
  "
30136 PRINT"▒
  "
30137 IFA$="Q"THEN30145
30140 NEXTR:GOTO20000
30145 IFLV=6THEN30150
30146 IFG=87THEN30140
30147 FP$="Y"
30150 PRINT"TTTT]":A=0
30160 IFLV<>6THEN610
30170 FP$="":FP=1:IFG=81THEN630
30180 GOTO640
31000 PRINT"]":END
```

........................Solitario

L'antico gioco del Solitario e' ....(non vi sembra un discorso gia' sentito?) ...stato fedelmente riprodotto in questo programma. Attualmente ci sono diversi tipi di solitari, che differiscono solo per la loro forma della scacchiera. Questo gioco viene effettuato su un campo triangolare, con a disposizione 15 caselle occupate da 14 pezzi. Lo scopo del gioco consiste nel catturare ogni pezzo, uno alla volta, rimuovendoli dalla scacchiera in modo tale da lasciare un pezzo soltanto, possibilmente rosso ed al centro della base del triangolo (ed e' qui che si riconosce il "professionista" del solitario).

Una cattura viene effettuata "saltando" un pezzo da una posizione adiacente verso una casella vuota; il programma permette al giocatore di scegliere inizialmente la posizione della casella vuota e della pedina rossa, o in alternativa di accettare un posizionamento casuale dei pezzi. Le posizioni sono numerate da 1 a 15 ed i numeri possono essere visualizzati o meno, a discrezione del giocatore.

Tutte le mosse sono da digitare nella forma "DA" seguito da "A" (due entrate separate, seguite ognuna da RETURN), e

il giocatore si dovra' arrendere quando non potra'fare altro che premere semplicemente RETURN.

```
3 REM *** PRG.#43- SOLITARIO ***
4 GOTO100
5 PRINT"▣PREMI SPAZIO QUANDO SEI PRONTO"
6 Q$="":GETQ$:IFQ$=""THEN6
7 RETURN
100 DIMB(15)
110 FORJ=1TO15:READP:B(J)=P:NEXT
120 DATA1442,1560,1564,1678,1682
130 DATA1686,1796,1800,1804,1808
140 DATA1914,1918,1922,1926,1930
150 D=54272
160 POKE53280,9
170 POKE53281,5
180 PRINTCHR$(5)
1000 PRINTCHR$(147);TAB(15)"SOLITARIO"
1010 PRINT"■■■■■■■■■■■■■■▣▣▣▣SCEGLI OPZION
I:-
1020 PRINT"▣▣1: RANDOM SET-UP
1030 PRINT"▣2: TUO SET-UP
1040 PRINT"▣3: ISTRUZIONI
1100 GOSUB6:IFQ$<"1"ORQ$>"3"THEN1100
1105 IFQ$="3"THEN7000
1110 DIMA(15)
1120 FORJ=1TO15:A(J)=1:NEXT
1130 IFQ$="2"THEN2000
1140 A=RND(0)
1150 A=INT(RND(1)*15)+1
1160 A(A)=2
1170 B=INT(RND(1)*15)+1
1180 IFB=ATHEN1170
1190 A(B)=0
1200 GOTO3000
2000 REM** ENTER SET UP **
2005 GOSUB2100
2010 INPUT"▣▣▣▣POSIZIONE ROSSO (1-15)*■■■
■■";A$
2020 A=VAL(A$)
2030 IFA<1ORA>15ORA<>INT(A)THENPRINT"TTT
TTT":GOTO2010
2040 A(A)=2
2050 PRINT"▣"
2060 INPUT"POSIZIONE VACANTE (1-15) *■■■■
";B$
2070 B=VAL(B$)
2080 IFB=AORB<1ORB>15ORB<>INT(B)THENPRIN
T"TT":GOTO2060
2085 A(B)=0
2090 GOTO3000
2100 PRINT"▣"TAB(18)"1"
2105 PRINT
2110 PRINTTAB(16)"2   3
2115 PRINT
2120 PRINTTAB(14)"4   5   6
2125 PRINT
2130 PRINTTAB(12)"7   8   9   10
2135 PRINT
2140 PRINTTAB(10)"11  12  ▣13■  14  15
2150 RETURN
3000 REM** GAME **
3005 PRINT"▣▣▣▣▣▣▣▣▣VOLETE I NUMERI DEL B
ORDO      (S OR N)?"
```

```
3010 GOSUB6:IFQ$="N"THENPRINT"▫":GOTO304
0
3020 IFQ$<>"S"THEN3010
3030 GOSUB2100
3040 GOSUB5000
3150 PRINT"▪▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫"
3155 PRINT"                    ▪":PRINT"
 ▫▫▫"
3160 INPUT"DA          F▪▪▪";A$
3170 IFA$="F"THEN10000
3180 A=VAL(A$)
3190 IFA<1ORA>15THENPRINT"▫▫":GOTO3160
3200 IFPEEK(B(A))<>87ANDPEEK(B(A))<>81TH
ENA$="0":GOTO3180
3205 PRINT
3210 INPUT" A          T▪▪▪";B$
3220 B=VAL(B$)
3230 IFB<1ORB>15THENPRINT"▫▫":GOTO3210
3240 IFPEEK(B(B))<>42THENGOSUB6000:GOTO3
150
3300 C=(B(A)+B(B))/2
3310 IFPEEK(C)<>81ANDPEEK(C)<>87THENGOSU
B6000:GOTO3150
3320 IFABS(C-B(A))>130THENGOSUB6000:GOTO
3150
3330 RY=7:IFPEEK(B(A))=81THENRY=2
3400 POKEC,42:POKEC+D,0
3410 POKEB(B),PEEK(B(A)):POKEB(B)+D,RY
3420 POKEB(A),42:POKEB(A)+D,0
3430 FORJ=1TO15
3440 IFC<>B(J)THENNEXT
3450 IFJ=16THENSTOP:REM** ERROR IF THIS
POINT REACHED **
3460 A(B)=A(A):A(A)=0:A(J)=0
3500 GOTO3150
5000 REM ** POKE BOARD **
5010 FORJ=1TO15
5020 IFA(J)=1THENPOKEB(J),87:POKEB(J)+D,
7
5030 IFA(J)=2THENPOKEB(J),81:POKEB(J)+D,
2
5040 IFA(J)=0THENPOKEB(J),42:POKEB(J)+D,
0
5050 NEXT
5060 RETURN
6000 REM ** ILLEGAL MOVE **
6010 PRINT"▪▫▪ MOSSA ILLEGALE"
6015 PRINT"▪▫▪  C PER PULIRE "
6020 GOSUB6:IFQ$<>"C"THEN6020
6030 PRINT"▪▫                             "
6035 PRINT"▪                             "
6040 RETURN
7000 REM** RULES **
7010 PRINTCHR$(147);TAB(17)"RULES"
7020 PRINTTAB(17);:FORJ=1TO5:PRINTCHR$(1
01);:NEXT:PRINT
7030 PRINTCHR$(144);CHR$(42)CHR$(5)" RAP
PRESENTA UNO SPAZIO VACANTE.
7040 PRINTCHR$(17);CHR$(158);CHR$(119)CH
R$(5)" RAPPRESENTA I PEZZI GIOCANTI.
7050 PRINTCHR$(17);CHR$(28);CHR$(209);CH
R$(5)" RAPPRESENTA I PEZZI ROSSI GIOCANT
I.
7060 PRINT"▪▫         MOSSA:- SALTA UN PEZZO
GIOCANTE FINO
7070 PRINT"AD UNO SPAZIO LIBERO, CANCELL
ANDO IL
```

```
7080 PRINT"PEZZO PRESO DALLO SCHERMO. RI
PETI CIO'
7090 PRINT"FINO A CHE NON RIMANGANO PEZZ
I.
7100 PRINT"PER UN MIGLIORE PUNTEGGIO,
DEVE RIMANERE
7110 PRINT"UN PEZZO ROSSO SUL NUM. 13.
7120 PRINT" PREMI SPAZIO PER TORNA
RE AL MENU"
7130 GOSUB6
7140 IFQ$=CHR$(32)THEN1000
7150 GOTO7130
10000 REM ** END **
10100 C=0:R=0:FORJ=1TO15
10110 IFA(J)<>0THENC=C+1
10115 IFA(J)=2THENR=1
10120 NEXT
10200 PRINT""C"SINISTRA"
10210 PRINT
10220 IFC>3THENPRINT"NON ABBATTERTI!"
10230 IFC=3THENPRINT"DISCRETO"
10240 IFC=2THENPRINT" BUONO"
10250 IFC=1ANDR=0THENPRINT"MOLTO BUONO"
10260 IFC=1ANDR=1ANDPEEK(B(13))<>81THENP
RINT"BRILLANTE"
10270 IFC=1ANDR=1ANDPEEK(B(13))=81THENPR
INT"GENIO !!!"
10280 PRINT"                    "
11000 PRINTTAB(22)"N PER UN NUOVO GIOCO
"
11010 PRINT
11020 PRINTTAB(25)"F PER FINIRE"
11100 GOSUB6:IFQ$="N"THENRUN
11110 IFQ$<>"F"THEN11100
11120 PRINT"":END
```

........................Bocce

Questo programma rappresenta una ragionevole simulazione del gioco delle bocce (o "boccette"), sebbene abbiate molte possibilita' di modificare il gioco, se lo desiderate. Il gioco e' per due giocatori (o uno che giochi entrambi le mani). Il "boccino" viene posizionato casualmente sulla destra del campo, mentre la boccia e' alla sinistra. Ogni giocatore ha a disposizione quattro bocce per avvicinarsi il piu' possibile al boccino.

Ogni giocatore, al suo turno, procede nel modo seguente:

Posiziona la boccia usando i tasti-cursore (non e' possibile andare nella parte destra dello schermo). Dopo aver trovato la giusta posizione, preme RETURN.

Decide l'effetto, importantissimo in questo gioco, grazie al comando BIAS: ogni boccia ha un contrappeso interno che non le fa seguire un traiettoria diritta. Si puo' scegliere l'intensita' dell'effetto (da 1 a 5) e la direzione (Destra o Sinistra). Bastera' premere D o S, seguito dal desiderato valore d'effetto.

Decide l'intensita' del colpo necessaria per raggiungere il boccino; i valori variano da 1 a 40, e una volta digitato il numero desiderato basta premere RETURN per lanciare la boccia.

Dopo che il primo giocatore ha realizzato il primo tiro, il gioco passa al secondo, che segue l'identica procedura, fino a che entrambi hanno terminato i quattro tentativi a loro disposizione.

Nel programma il boccino e le bocce (una volta lanciate) sono immobili. Se una boccia colpisce il boccino, quest'ultimo non si muove. Se una boccia tocca un'altra boccia precedentemente lanciata, essa viene fermata ed eliminata.

Ogni boccia che non supera la meta' del campo o tocca il bordo e' fuori gioco.

```
5 REM*** PRG.44-BOCCE ***
10 POKE53280,13:POKE53281,5:PRINTCHR$(14
7);CHR$(153)
20 TL=1024:D=54272
30 FORJ=TLTOTL+39:POKEJ,160:POKEJ+D,13:P
OKEJ+960,160:POKEJ+960+D,13:NEXT
40 FORJ=TL+40TOTL+79:POKEJ,160:POKEJ+880
,160:POKEJ+D,9:POKEJ+880+D,9:NEXT
50 FORJ=TL+79TOTL+919STEP40:POKEJ,160:PO
KEJ+1,160:POKEJ+D,9:POKEJ+1+D,9:NEXT
60 M=12:MP=TL+5+40*M
70 FORJ=1TO39:B$=B$+CHR$(32):NEXT:B$=B$+
CHR$(145)
80 X=RND(-TI)
100 X=INT(RND(1)*6):Y=INT(RND(1)*19):J=T
L+152+X+40*Y
110 POKEJ,81:POKEJ+D,1
120 FORB=1TO4:FORP=1TO2:LR$="0"+CHR$(32)
130 GOSUB2000
140 GOSUB1000
150 PRINT"TASTI CURSORE=MOV.LANC.: RETUR
N=PRONTO";
160 POKEMP,102:POKEMP+D,11
170 GETI$:IFI$=CHR$(13)THEN300
```

```
180  MN=40*((I$=CHR$(17))*(MP<TL+880))
190  MN=MN-40*((I$=CHR$(145))*(MP>TL+120)
)
200  MN=MN+((I$=CHR$(29))*((MP-TL)/40-INT
((MP-TL)/40)<.21))
210  MN=MN-((I$=CHR$(157))*((MP-TL)/40-IN
T((MP-TL)/40)>.051))
220  IFMN=0THEN170
230  POKEMP+D,5
240  MP=MP+MN
250  GOTO160
300  GOSUB1000
310  PRINT"BIAS:  S/D SEGUITO DA UN NUM.  (
1-5)";
320  L$="":GETL$:IFL$<>"S"ANDL$<>"D"THEN3
20
330  LR$=L$:GOSUB2000
340  R$="":GETR$:IFR$<"1"ORR$>"5"THEN330
350  LR=VAL(R$)
360  IFL$="S"THENLR=-LR
370  LR$=L$+R$:GOSUB2000
400  GOSUB1000
410  PRINT"FORZA DEL COLPO (1-40) POI   RE
TURN";
420  S$=""
430  I$="":GETI$:IFI$=CHR$(13)THEN500
440  IFI$<"0"ORI$>"9"THEN430
450  S$=S$+I$
460  GOTO430
500  S=VAL(S$)
510  IFS<1ORS>40THEN420
600  Q=MP+1:PQ=Q:L=0:X=0
610  FORK=QTOMP+S:Y=0
620  L=L+1:LQ=PEEK(Q+D)AND15
630  IFL>.5*SANDRND(1)>.2THENX=X+1:IFX=6-
ABS(LR)THENY=40:X=0
640  IFLQ=1THENQ=Q+40*SGN(LR)
650  IFLQ=9ORLQ=0ORLQ=2THENK=MP+S:IFLQ<>9
THEN690
660  POKEPQ+D,5
670  POKEQ,176+B:POKEQ+D,2*ABS(P=2)
680  PQ=Q:Q=Q+1+Y*SGN(LR)
690  NEXTK
700  NEXTP,B
800  GOSUB1000
810  PRINT"GAME OVER  :  PREMI SPAZIO PER R
IGIOCARE";
820  GETI$:IFI$<>CHR$(32)THEN820
830  RUN
1000 PRINTCHR$(19):FORJ=1TO23:PRINT:NEXT
1010 PRINTCHR$(18)B$:PRINTCHR$(18);
1020 RETURN
2000 PRINTCHR$(19);CHR$(18)"GIOCAT"P,"BO
CCE"B,"BIAS:";LR$
2010 RETURN
```

........................Cercaparole

Sono sicuro che abbiate gia' visto questo tipo di puzzle

su qualche rivista; una griglia 15x15 di lettere casuali contenete 10 parole disposte orizzontalmente, verticalmente o diagonalmente, diritte o al contrario.

Il programma disegna la scacchiera e posiziona il cursore a sinistra in alto. Usate i tasti-cursore per muovere il quadratino lungo la griglia, fino a trovare la prima lettera di una parola nascosta. Quindi premete il tasto corrispondente al numero della parola (0-9), e se avete posizionato il cursore correttamente, la parola cambiera' colore.

Se non riuscite a trovare altre parola, premendo RETURN vedrete visualizzarsi tutte le parole presenti sulla scacchiera in un colore diverso.

Le istruzioni DATA (linee 40-50) contengono le dieci parole nascoste, e potrete naturalmente cambiarle ogni volta che vorrete. State attenti, pero', che il programma non comprende parole piu' lunghe di 15 caratteri.

Le lettere casuali usate per riempire la griglia sono selezionate da una stringa (linea 290) che contiene tutte le lettere delle parole nascoste.

```
5 REM*** PRG.45-CERCAPAROLE ***
10 X=RND(-TI)
20 POKE53280,10:POKE53281,2:PRINTCHR$(14
7);CHR$(5)"PLEASE WAIT A FEW MOMENTS"
40 DATACOMMODORE,APPLE,SINCLAIR,ELECTRON
,ORIC
50 DATABASIC,FORTRAN,PASCAL,COBOL,FORTH
100 DIMW(14,14)
110 FORJ=0TO9:READA$(J)
120 L=LEN(A$(J))-1
130 SX=INT(RND(1)*15)
140 SY=INT(RND(1)*15)
150 DX=INT(RND(1)*3)-1
160 DY=INT(RND(1)*3)-1
170 IFDX=0ANDDY=0THEN150
180 FX=SX+(L*DX)
190 FY=SY+(L*DY)
200 IFFX<0ORFX>14ORFY<0ORFY>14THEN130
210 FORK=0TOL
220 IFW(SX+(DX*K),SY+(DY*K))=ASC(MID$(A$
(J),K+1,1))THEN240
230 IFW(SX+(DX*K),SY+(DY*K))<>0THEN130
240 NEXT
250 S(J,1)=SX
260 S(J,2)=SY
```

```
270  D(J,1)=DX
280  D(J,2)=DY
290  R$=R$+A$(J)
300  FORK=0TOL
310  W(SX+(DX*K),SY+(DY*K))=ASC(MID$(A$(J
),K+1,1))
320  NEXTK,J
400  FORJ=0TO14:FORK=0TO14
410  R=INT(RND(1)*LEN(R$))+1
420  IFW(J,K)=0THENW(J,K)=ASC(MID$(R$,R,1
))
430  NEXTK,J
500  POKE53280,8:PRINTCHR$(147)
510  FORJ=0TO14:PRINTTAB(8);
520  FORK=0TO14
530  PRINTCHR$(W(J,K));
540  NEXTK
550  IFJ<10THENPRINTSPC(3)J;A$(J);
560  PRINT:NEXTJ
570  PRINT:PRINT"USA I TASTI CURSORI PER
MUOVERTI  "
580  PRINT:PRINT"PREMI NUMERI (0-9) QUAND
O TROVI L'I-     NIZIO DI UNA PAROLA"
590  PRINT:PRINT"PREMI 'RETURN' PER RINUN
CIARE"
600  CX=0:CY=0:TI$="000000"
610  GETI$:PRINTCHR$(19)TAB(29)TI$
620  CP=1072+CX+40*CY
630  POKECP,PEEK(CP)+128
640  IFI$=CHR$(13)ORSC=10THEN1000
650  IFI$>="0"ANDI$<="9"THENGOSUB800
660  POKECP,PEEK(CP)-128
670  CX=CX+((I$=CHR$(29))*(CX<14))-((I$=C
HR$(157))*(CX>0))
680  CY=CY+((I$=CHR$(17))*(CY<14))-((I$=C
HR$(145))*(CY>0))
690  GOTO610
800  V=VAL(I$):C=7
810  IFCX=S(V,2)ANDCY=S(V,1)ANDA$(V)<>""T
HENGOSUB2000:SC=SC+1:A$(V)=""
820  RETURN
1000  POKECP,PEEK(CP)-128
1010  FORV=0TO9:C=0
1020  IFA$(V)<>""THENGOSUB2000
1030  NEXT
1040  PRINTCHR$(19)"PUNTEGGIO  "SC
1050  FORJ=1TO20:PRINT:NEXT:END
2000  P=55344+40*S(V,1)+S(V,2)
2010  FORK=0TOLEN(A$(V))-1
2020  POKEP,C
2030  P=P+40*D(V,1)+D(V,2)
2040  POKE55365+40*V+K,C
2050  NEXT
2060  RETURN
```

E ancora !

..........................Loony lander

..........................Skyline

..........................Ponte pontone

A beneficio degli illetterati, un "pontone" e' una struttura temporanea costruita sopra un fiume udando zattere e tronchi.
Voi ricoprite il ruolo di un pilota d'areoplano, e la vostra missione consiste nel far cadere le zattere lungo il fiume a formare il pontone. Un aeroplano amico cerchera' di aiutarvi e dovrete ad ogni costo evitare di colpirlo direttamente con una zattera o accatastando piu' di tre zattere una sull'altra, cosa che lo farebbe precipitare.

Il punteggio e': 1 punto per ogni zattera lanciata nel fiume, e -1 punto per ogni zattera lanciata su un'altra zattera. Alla fine del gioco, quando o il ponte e' completo, o avrete incidentalmente distrutto l'aereo amico, la complessa formula alla linea 900 vi rivelera' il vostro punteggio.

..........................Note del programma

Le linee 60 e 70 possono essere alterate variando l'altezza alla quale i due areoplani volano sopra il fiume. Aumentando B si evitera' di colpire facilmente tramite le zattere accatastate l'areoplano amico che vola a bassa quota.

I due areoplani volano in direzioni opposte (linee 240 e 310), ma se volassero alla stessa velocita' si incontrerebbero sempre nello stesso punto dello schermo.

Per questo, la linea 270 provvede affinche' l'aereo piu' basso voli approssimativamente ad una velocita' piu' bassa del 10% rispetto all'altra.

```
5 REM *** PRG.46- PONTE PONTONE ***
10 POKE53280,14
20 POKE53281,3
30 D=54272
40 PRINTCHR$(144);CHR$(147)
50 FORJ=1944TO2023:POKEJ,160:POKEJ+D,6:N
EXT
60 A=10:REM HEIGHT OF UPPER PLANE
70 B=3: REM HEIGHT OF LOWER PLANE
80 P1=1944-A*40:P2=P1+40
90 Q1=1983-B*40:Q2=Q1-40
100 P=P1:Q=Q1
110 R=RND(-TI)
120 POKE198,0
130 TI$="000000"
200 PRINTCHR$(19)TI$,"SCORE"S;CHR$(157);
CHR$(32):IFT=40THEN900
210 POKEP,32
220 IFF=0THENX=P+40
230 GETI$:IFI$<>""THENF=1:NH=137:NL=47:W
=17:GOSUB1000
240 P=P+1:IFP=P2THENP=P1
250 POKEP,62:POKEP+D,0
260 IFF=1THENGOSUB500:IFL=1THEN900
270 IFRND(1)>.9THEN200
300 POKEQ,32:POKEQ+D,3
310 Q=Q-1:IFQ=Q2THENQ=Q1
315 IFPEEK(Q)=230THENX=Q:GOSUB600:GOTO90
0
320 POKEQ,81:POKEQ+D,2
330 GOTO200
500 X=X+40
510 Y=PEEK(X)
520 IFY=81THENGOSUB600:RETURN: REM HIT L
OWER PLANE
530 IFY=230THENGOSUB700:RETURN: REM HIT
OTHER RAFT
540 IFY=160THENGOSUB800: REM HIT SEA
550 POKEX-40,32:POKEX-40+D,3
560 POKEX,230:POKEX+D,4
570 RETURN
600 NH=45:NL=198:W=129
610 GOSUB1000
620 FORJ=1TO250:POKEX+D,INT(RND(1)*15):N
EXT
630 GOSUB2000
640 L=1
650 RETURN
700 S=S-1
710 NH=17:NL=37:W=33
720 GOSUB1000:GOSUB2000
730 F=0
740 RETURN
800 S=S+1:T=T+1
810 NH=45:NL=198:W=33
820 GOSUB1000:GOSUB2000
830 F=0
840 RETURN
```

```
900 R=(INT(S↑3/VAL(TI$)*100)/100)-(500*(
T=40))
910 PRINTCHR$(17)"▩GAME OVER","■TUA CLAS
SIFICAZIONE"R
920 PRINTCHR$(17)"PREMI ▩SPAZIO■ PER UN'
ALTRA PARTITA"
930 GETI$:IFI$<>CHR$(32)THEN930
940 RUN
1000 POKE54296,15
1010 POKE54277,200
1020 POKE54278,100
1030 POKE54273,NH
1040 POKE54272,NL
1050 POKE54276,W
1060 RETURN
2000 POKE54276,0
2010 POKE54277,0
2020 POKE54278,0
2030 RETURN
```

...............Anti-aircraft

Sebbene sia noto a tutti che un movimento grafico scritto in BASIC non sia affatto veloce, questo programma crea l'illusione di una tale velocita' da veder muovere uno schermo intero alla volta.

L'obbiettivo e' colpire quanti piu' areoplani nemici e' possibile in due minuti, usando il vostro cannone anti-aircraft da una base terrestre. Come sempre, il tasto-cursore verticale vi muovera' verso sinistra, quello orizzontale verso destra, mentre il tasto RETURN vi fara' sparare

Voi potete "far fuori" molti (ma non tutti) gli aerei nemici, ma non potete sparare se il colpo precedente non ha ancora raggiunto l'obbiettivo o il bordo dello schermo. Il nemico vi lancia delle bombe costantemente angolate verso la vostra postazione, in modo da costringervi ad un continuo movimento.

E per rendere le cose ancora un po' piu' complicate, il vostro nemico sara' in grado ogni tanto di rendersi invisibile.

.......................Note del programma

Il segreto che permette l'illusione della velocita' in questo gioco e' consiste nel fatto che l'azione principale si svolqe nella meta' dello schermo inferiore; in questo modo le bombe nemiche sembrano viaggiare piu' velocemente, e sono difficili da evitare. Las parte alta del video e' costellata casualmente di stelle (linea 140), in modo tale che i vostri colpi a vuoto dovranno incontrare il bordo superiore dello schermo per potervi ridare la possibilita' di sparare nuovamente. I punti alle linee 830 e 835 sono li' semplicemente per riposizionare i punti nascosti dai messaggi di gioco finito, e la loro posizione non e' importante. Gli spazi alle linee 3100-3550 servono a cancellare i vari messaggi di istruzione; potete evitare questa procedura usando CHR$(147), cancellando l'intero schermo dopo ogni gioco.

```
5 REM*** PRG.#47-ANTI-AIRCRAFT ***
10 POKE53280,6:POKE53281,6
20 D=54272
30 PRINTCHR$(147);CHR$(159)
40 TL=1024:TR=TL+39:BL=TL+(40*22):BR=BL+
39
50 GOSUB3000
100 POKETL,112:POKETR,110:POKEBL,109:POK
EBR,125
110 FORJ=1TO38:POKETL+J,64:POKEBL+J,64
120 IFJ<22THENPOKETL+40*J,93:POKETR+40*J
,93
130 NEXT
140 FORJ=1TO100:P=RND(1)*400+TL:IFPEEK(P
)=32THENPOKEP,46
150 NEXT
160 TI$="000000"
170 S=0:S1=0:H=0:H1=0
200 MX=20:MY=1
210 M=BL+MX-40*MY
230 TX=0:TY=INT(RND(1)*8)+4
240 COL=INT(RND(1)*8)
250 TM=1:IFRND(1)>.9THENTM=2
300 POKEM,102:POKEM+D,10
310 GOSUB500
330 FORJ=1TO38STEPTM
340 PRINTTAB(33);MID$(TI$,4,1)"."MID$(TI
$,5);"]":TT$=TI$
350 IFTT$>="000200"THENPRINT"[illegible]
GAME OVER":GOTO800
360 TX=TX+TM:IFTX>38THENTX=38
370 T=BL+TX-40*TY
380 POKET,81:POKET+D,COL
390 GOSUB600
400 IFBF=0THENBF=1
```

```
410  IFBF>0THENGOSUB700
420  POKET,32:POKET+D,3
430  IFH1>HTHENH=H1:GOSUB500:POKEM,102:PO
KEM+D,10
440  GOSUB600
450  IFS1>STHENS=S1:GOTO230
460  NEXT
470  IFBF>1THENGOSUB700:GOTO470
480  GOTO230
500  PRINT"SUCCES
SI:VOI";S,"THEM";H;
510  RETURN
600  Q=PEEK(197):GOSUB900
605  IFMF<1THEN665
610  IFMF=1THENMF=2:MP=M-40:GOTO660
615  POKEMP,32
620  MP=MP-40:IFPEEK(MP)=32THEN660
630  GOSUB2500
640  IFPEEK(MP)=81THENFORK=1TO3:GOSUB2000
:NEXT
645  IFPEEK(MP)=126THENPOKEMP,42:FORK=1TO
15:NEXT:POKEMP,32:F=0:GOTO655
650  POKEMP+40,42:FORK=1TO10:NEXT:POKEMP+
40,32
655  GOSUB2600:MF=0:GOTO665
660  POKEMP,30
665  RETURN
700  IFBF=1THENBF=2:DX=TX-MX:DY=TY-MY:BP=
T+40-SGN(DX):GOTO735
705  POKEBP,32
710  BP=BP+40-SGN(DX):Z=PEEK(BP):IFZ=32TH
EN735
715  GOSUB2500
720  EP=BP-40+SGN(DX):IFZ=102THENEP=BP-40
725  IFZ=102THENPOKEBP,98:H1=H+1:FORK=1TO
5:GOSUB1005:NEXT:POKEBP,32
730  GOSUB1000:GOSUB2600:RETURN
735  POKEBP,126:RETURN
800  IFMF>0THENMF=0:POKEMP,32
805  IFBF>0THENBF=0:POKEBP,32
810  IFS=HTHEN810
815  A$="VOI":IFH>STHENA$="LORO"
820  PRINT""A$"  WIN
BY";ABS(S-H)
825  FORJ=1TO2000:NEXT
830  PRINT"                     -  -
      ":S=0:H=0:POKEM,32
835  PRINT"                     -        -
 -       -   -                  "
840  SR=1:GOSUB3045
845  GOTO160
900  IFQ=7ANDMX>1THENPOKEM,32:M=M-1:MX=MX
-1:POKEM,102:POKEM+D,10
905  IFQ=2ANDMX<38THENPOKEM,32:M=M+1:MX=M
X+1:POKEM,102:POKEM+D,10
910  IFQ=1ANDMF=2ANDMP<BL-480THENPOKEMP,4
2:POKEMP,90:POKEMP,32
915  IFQ=1ANDMF=0THENMF=1
920  RETURN
1000  BF=0
1005  POKEEP,42
1010  POKEEP-40,42
1015  X1=PEEK(EP-41):POKEEP-41,42
1020  X2=PEEK(EP-39):POKEEP-39,42
1025  POKEEP,32
1030  POKEEP-40,32
1035  POKEEP-41,X1
```

```
1040 POKEEP-39,X2
1045 RETURN
2000 EP=MP
2005 IFEP-80>TLTHENX3=PEEK(EP-80):POKEEP
-80,42
2010 GOSUB1005
2015 IFEP-80>TLTHENPOKEEP-80,X3
2020 S1=S+1
2030 RETURN
2500 POKE54296,15
2510 POKE54277,200
2520 POKE54278,200
2530 POKE54273,137
2540 POKE54272,43
2550 POKE54276,129
2560 RETURN
2600 POKE54276,0
2610 POKE54277,0
2620 POKE54278,0
2630 RETURN
3000 PRINT"ANTI-AIRCRAF
T."
3010 PRINT"USA CURSOR SU/GIU' PER
LA SINISTRA"
3015 PRINT"USA CURSOR SIN./DEST. PE
R LA DESTRA"
3020 PRINT"USA RETURN PER SPARARE."
3030 FORJ=55296TO56295:POKEJ,3:NEXT
3040 PRINT"HAI DUE MINUTI DI
TEMPO"
3045 PRINT"PREMI SPAZ
IO QUANDO PRONTO"
3050 GETQ$:IFQ$<>CHR$(32)THEN3050
3055 IFSR<>1THEN3500
3100 PRINT"
            "
3110 RETURN
3500 PRINT"
            "
3510 PRINT"
            "
3520 PRINT"
          "
3530 PRINT"
 "
3540 PRINT"
            "
3550 PRINT"
                 "
3560 RETURN
```

............................Loony lander

Questo programma fu scritto originariamente per il Commodore Pet 4032 (il nonno del 64), e non ho fatto altro che le strette indispensabili modifiche per permettere al programma di girare sul 64. L'indirizzo di schermo e'

stato cambiato (linea 5) e i colori aggiunti (tutti i numeri di linea che finiscono con 3 e con 7); per il resto il programma e' identico alla versione per il Pet. Lo scopo per cui ho incluso questo programma e' principalmente quello di farvi rendere conto che il Commodore 64 non e' l'unico computer al mondo. C'e' una enorme famiglia di computer, ed esistono migliaia di programi per macchine dal VIC a PET, pronti ad un qualsiasi tentativo di adattamento e di modifica. Cercate sulle riviste per avere qualche idea.

Come avrete capito dal titolo, un modello piu' povero di "Lunar lander", il concetto di questo gioco non sara' mai serio totalmente. Infatti questo gioco e' stato scritto come un "anti-game" ed include molte delle caratteristiche che si trovano nei giochi d'arcade peggiori. Sfortunatamente il gioco non lavora, sembra fermo. Ammettete che l'idea di far atterrare la vostra astronave sul terreno sgomberato dalle macerie grazie alle vostre bombe e' un po' sciocca...

Usate la barra spaziatrice per lanciare una bomba, e SHIFT+Spazio per sparare un missile. I missili sono da considerarsi come ultima risorsa, e sono sparati orizzontalmente dalla navicella: costano 20 punti ad ogni lancio, e non possono essere usati se si possiedono meno di venti punti. Viene calcolato un punto per ogni blocco di macerie distrutto con una bomba.

Un terzo, certamente inferiore, metodo per rimuovere le macerie consiste nel precipitare con l'astronave su di un blocco. Questo vi evitera' delle penalita' ma non ve lo raccomando troppo come metodo poiche' avete soltanto cinque navicelle a disposizioni con le quali effettuare un atterraggio riuscito (nella parte bassa dello schermo, a destra).

```
1 REM*** PRG.#48- LOONY LANDER ***
3 POKE53280,3:POKE53281,0:PRINTCHR$(5)
5 P=1024:D=54272
```

```
10 Q1=41:Q2=42:Q3=43:Q4=44:Q5=45:QB=83
15 PRINTCHR$(147)"SCORE=";:SC=0:PL=5
20 FORJ=480TO999STEP2:POKEP+J,247:POKEP+
D+J,INT(RND(1)*15)+1
25 J=J-(RND(1)*J<500)
30 NEXT
35 BF=0
40 Q=P+80*(5-PL)
43 POKEQ+D+Q1,2:POKEQ+D+Q2,2:POKEQ+D+Q3,
2:POKEQ+D+Q4,2
45 POKEQ+Q1,91:POKEQ+Q2,64:POKEQ+Q3,182-
PL:POKEQ+Q4,91:POKEQ+Q5,64
47 POKEQ+D+Q5,2
50 PRINTSC;"■■  ■";TAB(6);
55 IFBF=0THENBP=Q+QB
60 GETQ$:IFQ$=CHR$(32)THENBF=1
65 IFQ$=CHR$(160)ANDSC>20THENGOSUB2000
70 IFBF=1THENGOSUB1000
75 POKEQ+Q1,32:Q=Q+1
80 IFPEEK(Q+Q5)=247THEN6000
85 IFQ+Q5=P+999THEN7000
90 GOTO45
1000 POKEBP,81:POKEBP+D,7:BN=BP+40
1010 IFPEEK(BN)<>247THEN1020
1013 POKEBN+D,8
1015 POKEBN,42:POKEBN,163:POKEBN,42:POKE
BN,32:POKEBP,32:BP=Q+QB:BF=0:SC=SC+1
1017 POKEBN+D,0
1020 POKEBP,32:POKEBP+D,0:BP=BN
1030 IFBP>P+999THENBF=0
1040 RETURN
2000 MP=Q+Q5+1
2010 IFPEEK(MP)=247ORMP>P+999THEN2100
2015 IFMP>P+999THEN2100
2020 POKEMP,81:POKEMP+D,8
2030 POKEMP,32
2040 MP=MP+1:GOTO2010
2100 POKEMP,42:POKEMP,32:POKEMP,42:POKEM
P,32:POKEMP+D,0
2110 SC=SC-20
2120 RETURN
6000 FORK=1TO5
6005 POKEQ+Q1,42:POKEQ+Q2,42:POKEQ+Q3,42
:POKEQ+Q4,42:POKEQ+Q5,42
6010 POKEQ+Q2,163:POKEQ+Q3,163:POKEQ+Q4,
163
6015 POKEQ+Q1,163:POKEQ+Q2,32:POKEQ+Q3,3
2:POKEQ+Q4,32:POKEQ+Q5,163
6020 POKEQ+Q1,42:POKEQ+Q2,163:POKEQ+Q3,1
63:POKEQ+Q4,163:POKEQ+Q5,42
6025 POKEQ+Q2,42:POKEQ+Q3,42:POKEQ+Q4,42
6030 POKEQ+Q1,32:POKEQ+Q2,32:POKEQ+Q3,32
:POKEQ+Q4,32:POKEQ+Q5,32
6200 NEXT
6210 PL=PL-1:IFPL>0THEN35
6220 PRINT"■■■HAI FALLITO! I 5 PIANETI P
ERDUTI PER   ■";SC;"PUNTI !"
6230 IFSC<20THENPRINT"■(PATETICO!)
6240 GOTO7020
7000 PRINT"■■■BEN FATTO! NUMERO DI ATTER
RAGGI   ";6-PL
7010 PRINT"■CON UNO SCORE DI";SC
7020 PRINT"■■R▬RE-RUN  :  ■E▬END"
7030 GETQ$
7040 IFQ$="R"THENRUN
7050 IFQ$="E"THENPRINT"▯":END
7060 GOTO7030
```

.........................Skyline

Questo e' "Loony lander", completamente rivisto e dedicato al Commodore 64, con sprites, suono e caratteri grafici definiti.

Lo scenario del gioco e' stato un po' modificato: i blocchi di macerie ora sono dei grattacieli (l'idea di crearvi una pista d'atterraggio distruggendo dei palazzi e' ancora piu' ridicola della precedente). I missili sono stati del tutto eliminati, ed in ogni caso il programma risulta gia' essere lungo il doppio del precedente. Usate la barra spaziatrice per il lancio delle bombe.

La linea 1080 e' intesa a controllare quando avete vinto il gioco. Spero che prima o poi lavori. A me non e' mai successo di riuscire a distruggere tutti i palazzi.

.....................Note del programma

Le istruzioni REM di questo programma saranno sufficienti a guidarvi attraversole sue parti principali. Vi vorrei pregare di leggere (se non l'avete gia' fatto) i capitoli inerenti a "Generatore di caratteri", "Sprite-schermo" e "And/Or" PRIMA di digitare il programma Skyline.

Il carattere "@" (PEEK/POKE codice=0) e' il primo del set di caratteri, e per questo viene definito dagli indirizzi dal 12288 al 12295, una volta che il set di caratteri sia stato trasferito dalla ROM alla RAM (linee 10-70). La linea 80 ridefinisce questo carattere come un elemento dei grattacieli, usando i DATA alla linea 9000. E' da notare l fatto che, sebbene un solo carattere sia stato ridefinito, siamo stati costretti a trasferire tutto il set di caratteri grafici dalla ROM alla RAM, impedendoci dopo il RUN di listare il programma o di usare i normali caratteri alfa-numerici.

La linea 7 vi consente di far ripartire il programma anche dopo aver provocato uno STOP senza trasferire il set dei caratteri ogni volta. La linea 20040 inoltre fa ripartire il programma alla fine di ogni gioco senza il processo di trasferimento del tempo impiegato.

I due sprites sono definiti alle linee 100-150, usando i DATA alle linee 10000-10130. Gli spazi in queste istruzioni DATA non sono importanti, e sono state incluse per permettervi di digitare correttamente le linee DATA. Ogni linea orizzontale di ogni sprite richiede 3 bytes di data, cosi' dovreste trovare conveniente digitare le istruzioni a gruppi di tre, cosi' come ho fatto io.
Notate che, sebbene uno sprite possa avere una dimensione massima di 24x21 bits, non e' sempre necessario creare un oggetto cosi' grande. Lo sprite-bomba, per esempio, non supera gli otto bits; per questo e' importante tener conto dell'area dello sprite non usata quando si calcola la sua posizione.

La subroutine 6000 e' la parte del programma che calcola le posizioni sullo schermo per rispettare le precedenze degli sprites. La posizione calcolata sullo schermo e' sempre quella in alto a sinistra del video, anche se, nel caso della bomba, nessuno degli stessi oggetti si trova in quella posizione. La linea 6060 converte le coordinate x e y in indirizzi di schermo e calcola la visione dello sprite dal bordo in alto a siniistra fino alla punta del razzo (+42 bytes di schermo, quando SP=0) o all'estremita' della bomba (+81 bytes di schermo, quando SP=1).

Per il modo in cui il programma e' stato scritto, e' impossibile calcolare la collisione tra due sprites (il razzo e la bomba non vengono in collisione tra loro, ma passano per lo stesso punto ignorandosi). La collisione tra entrambi gli sprites ed un carattere, viene controllata dalla PEEK(53279), e dalla copertura di questo valore tramite l'uso di AND (linea 1090), che determina anche il tipo di sprite interessato.

```
3 REM*** PRG.#49- SKYLINE ***
5 POKE53280,3:POKE53281,6:PRINTCHR$(147)
;CHR$(5);
7 IFPEEK(52)=48ANDPEEK(56)=48THEN80
10 POKE52,48:POKE56,48
20 POKE56334,PEEK(56334)AND254
30 POKE1,PEEK(1)AND251
40 FORJ=0TO511:POKE12288+J,PEEK(53248+J)
:NEXT
50 POKE1,PEEK(1)OR4
60 POKE56334,PEEK(56334)OR1
70 POKE53272,(PEEK(53272)AND240)+12
80 FORJ=12288TO12295:READQ:POKEJ,Q:NEXT
90 D=54272
100 S(0)=704:S(1)=832
110 FORS=0TO1:FORJ=0TO62
120 READQ:POKES(S)+J,Q
130 NEXTJ,S
140 POKE2040,S(0)/64
150 POKE2041,S(1)/64
160 POKE53269,0: REM SPRITES OFF
200 POKE53287,7:REM COLOUR SPRITE 0
210 POKE53288,5:REM COLOUR SPRITE 1
220 POKE53279,0
300 S=0:L=0:GOSUB7000
500 DIMT(39)
510 T=RND(-TI)
520 FORJ=1TO12
530 FORK=1TO9:T(INT(RND(1)*40))=1:NEXTK
540 FORK=0TO39
550 IFT(K)=1THENPOKE1504+40*J+K,0:POKE15
04+D+40*J+K,INT(RND(1)*8)+8
560 NEXTK,J
999 REM : MAIN LOOP (ROCKET SPRITE)
1000 P=72:POKE53248,0:POKE53249,P:POKE53
264,0:POKE53269,1: REM ROCKET SPRITE ON
1010 X=0:Y=P
1020 X=X+8:IFX>=255THENX=0:POKE53264,PEE
K(53264)+1
1030 IFX>=85AND((PEEK(53264)AND1)=1)THEN
POKE53264,PEEK(53264)AND2:X=0:Y=Y+8
1040 POKE53248,X:POKE53249,Y
1050 GETI$
1060 IFI$<>""ANDPEEK(53269)=1THENGOSUB20
00
1070 IFPEEK(53269)=3THENGOSUB3000
1080 IFX=80ANDY=240AND((PEEK(53264)AND1)
=1)THEN20000
1090 IF(PEEK(53279)AND1)<>1THEN1020
1099 : REM : ROCKET HITS BUILDING
1100 L=L+1:GOSUB7000:POKE53287,2
1110 SP=0:GOSUB6000
1120 P=P+8
1130 POKE53287,7:POKE53248,0:POKE53249,P
1140 POKE53264,PEEK(53264)AND2
1150 IFL=5THEN20000
1160 POKE53279,PEEK(53279)AND2
1170 GOTO1010
1999 REM : SWITCH ON BOMB SPRITE
2000 NH=137:NL=43:GOSUB6500
2010 POKE53250,PEEK(53248)
2020 POKE53251,PEEK(53249)
2030 POKE53264,PEEK(53264)*3
2040 POKE53269,3
2050 GOSUB6570
2060 RETURN
2999 REM: MOVE BOMB SPRITE
```

```
3000  IFPEEK(53279)=2THEN4000
3010  IFPEEK(53251)<248THENPOKE53251,PEEK
(53251)+8:RETURN
3029  : REM : BOMB HITS GROUND
3030  NH=8:NL=147:GOSUB6500
3040  GOSUB5000
3050  GOSUB6570:RETURN
3999  REM: BOMB HITS BUILDING
4000  POKE53288,2:SP=1:GOSUB6000
4010  GOSUB5000:RETURN
4999  : REM : BOMB SPRITE OFF
5000  POKE53269,1:POKE53288,5
5010  POKE53264,PEEK(53264)AND1
5020  RETURN
5999  : REM : CALC SCREEN POSITION
6000  NH=22:NL=227:GOSUB6500
6010  SX=PEEK(53248+2*SP):IF(PEEK(53264)A
ND(SP+1))=SP+1THENSX=SX+256
6020  SY=PEEK(53249+2*SP)
6030  SX=INT((SX-24)/8)
6040  SY=INT((SY-50)/8)
6060  SZ=1024+SX+40*SY+42-39*(SP=1):IFSZ<
1024ORSZ>2023THEN6090
6070  IFPEEK(SZ)=0THENS=S+1:GOSUB7000
6080  POKESZ,32
6090  IFSP=0THENFORZ=1TO3:NH=NH+Z:GOSUB65
00:NEXTZ:GOSUB6570
6100  GOSUB6570:RETURN
6499  : REM : EXPLOSION (?)
6500  POKE54296,15
6510  POKE54277,33
6520  POKE54278,240
6530  POKE54273,NH
6540  POKE54272,NL
6550  POKE54276,17
6560  RETURN
6570  POKE54276,0
6580  POKE54277,0
6590  POKE54278,0
6600  RETURN
6999  : REM : SCORE
7000  PRINTCHR$(19)"SCORE"S:PRINT"RAZZI P
ERDUTI";L
7010  RETURN
8999  :     REM : BUILDING BLOCK
9000  DATA255,255,195,195,195,195,255,255
9999  :     REM : ROCKET
10000 DATA255,0,0, 255,240,0, 15,255,0,
15,255,240, 254,73,63, 254,73,63
10010 DATA15,255,240, 15,255,0, 255,240,
0, 255,0,0
10020 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
10099 :     REM : BOMB
10100 DATA0,195,0, 0,195,0, 0,126,0, 0,6
0,0, 0,24,0, 0,24,0, 0,60,0
10110 DATA0,126,0, 0,255,0, 0,255,0, 0,2
55,0, 0,126,0, 0,126,0
10120 DATA0,60,0, 0,60,0, 0,24,0, 0,24,0
10130 DATA0,0,0,0,0,0,0,0,0,0,0,0
19999 : REM : END
20000 H=100:NL=150:FORJ=1TO50:NH=H+J:GOS
UB6500:NEXT:GOSUB6570
20010 PRINT:PRINT:PRINTTAB(7)"PREMI UN T
ASTO PER RIGIOCARE"
20020 I$="":GETI$:IFI$=""THEN20020
20030 PRINTCHR$(147)
20040 RUN80
```

..................Comandi e Istruzioni PILOT

COMANDI (non possono essere usati nelle linee di programma)

NEW : Cancella il programma PILOT
RUN : Fa partire il programma PILOT
LIST : Lista sullo schermo il prog. PILOT
LLIST: Lista sulla stampante il prog. PILOT
LOAD : Carica in memoria un prog. PILOT da disco o nastro
SAVE : Salva su nastro o disco un prog. PILOT
BASIC: Ritorna al BASIC
EDITn: (n=num.linea) lista i numeri di linea, li cancella o li re-inserisce

(Tutti i comandi possono essere digitati totalmente, in parte o solo la prima lettera e SHIFT+seconda lettera)

ISTRUZIONI (possono essere usate solo all'interno di un programma)

Simbolo	seguito da:	Seguito da:
T	;(o condizione)	testo
J	;(o condizione)	testo
C	;	cod-colore
M	;	una risposta predeterminata
A	;	niente
*	;(qualsiasi cosa)	qualsiasi cosa

Nome:	Significato:
TYPE	Stampa il testo specificato sullo schermo.
JUMP	Salta ad un determinato numero di linea.

CLEAR SCREEN	Pulisce lo schermo con un determinato colore.
MATCH	Definisce la corretta risposta predeterminata.
ANSWER	Attende una risposta dall'utente.
REMARK	E' ignorato dall'interprete PILOT.

NOTA: Per la lista dei codici-colore, vedere programma "COLORE DELLO SFONDO".

CONDIZIONI (devono seguire un'istruzione T o J, al posto del punto e virgola).

Y : Esegue l'istruzione solo se la piu' recente risposta e' quella predeterminata.
N : Esegue l'istruzione solo se la risposta e' diversa da quella predeterminata.

I programmatori esperti troveranno in questo libro, programmi che illustrano la manipolazione di Bits e Bytes e la gestione dei Bits nell'uso di AND e OR.
Per i programmatori meno esperti è riportato un completo interprete PILOT, che permette la programmazione del C64 in un linguaggio ancora più semplice del BASIC.
Vi sono ancora programmi che illustrano e facilitano l'uso degli SPRITES, dei caratteri ridefiniti e del suono; viene inoltre proposto il programma "CARD BOX" un completo ed esauriente DATA BASE.
Per i principianti c'è un breve capitolo che introduce le semplici ed elementari caratteristiche del C64 e per gli stravaganti c'è un capitolo con mille sorprese entusiasmanti.
Dopo tutto ciò, potreste pensare che la parte principale di questo libro sia costituita dai programmi. Sono certamente importanti!
È però consigliabile che non li digitiate ciecamente ma vi soffermiate a chiedervi come e perché una routine esegue un determinato lavoro.
Con questo libro e con la cassetta in dotazione, nella quale sono memorizzati buona parte dei 50 programmi, troverete una risposta ad ogni vostro interrogativo. Infatti lo scopo principale di questo libro è di mostrarvi come lavorano i programmi, insegnandovi molti segreti sulla programmazione del COMMODORE 64.

Cod. 9150

L. 25.000
prezzo comprensivo di cassetta